

Jc859 U.S. PTO  
09/605461  
06/28/00

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application:

1999年 6月30日

出 願 番 号  
Application Number:

平成11年特許願第186491号

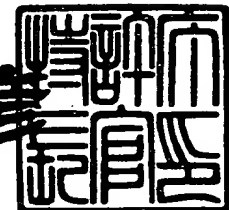
出 願 人  
Applicant(s):

株式会社次世代情報放送システム研究所  
ソニー株式会社

2000年 3月31日

特許庁長官  
Commissioner,  
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3022635

【書類名】 特許願

【整理番号】 9900310506

【提出日】 平成11年 6月30日

【あて先】 特許庁長官 伊佐山 建志 殿

【国際特許分類】 H04L 12/00  
H04L 29/00

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
内

【氏名】 山岸 靖明

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
内

【氏名】 権野 善久

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
内

【氏名】 西尾 郁彦

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
内

【氏名】 高林 和彦

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
内

【氏名】 原岡 和生

【特許出願人】

【識別番号】 597136766

【氏名又は名称】 株式会社次世代情報放送システム研究所

【代表者】 山田 敏之

【特許出願人】

【識別番号】 000002185

【氏名又は名称】 ソニー株式会社

【代表者】 出井 伸之

【代理人】

【識別番号】 100082762

【弁理士】

【氏名又は名称】 杉浦 正知

【電話番号】 03-3980-0339

【手数料の表示】

【予納台帳番号】 043812

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9708843

【包括委任状番号】 9904899

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 送信装置および送信方法、受信装置および受信方法、ならびに、送受信システムおよび送受信方法

【特許請求の範囲】

【請求項 1】 コンテンツの所在を階層的に管理するディレクトリの階層構造を送信する送信装置において、

自分の配下の情報を格納可能なコンテナエントリと、上記コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理する管理手段と、

上記管理手段に管理される上記ディレクトリの階層構造の変化を検出し、該検出の結果に基づき、上記コンテナエントリの変化の差分からなる第 1 の差分情報と、上記リーフエントリの変化の差分からなる第 2 の差分情報とをそれぞれ求める検出手段と、

上記第 1 の差分情報と上記第 2 の差分情報とを分離して送信する送信手段とを有することを特徴とする送信装置。

【請求項 2】 請求項 1 に記載の送信装置において、

上記送信手段は、上記コンテナエントリのそれぞれを識別する識別情報を上記第 1 の差分情報に付加して送信することを特徴とする送信装置。

【請求項 3】 請求項 1 に記載の送信装置において、

上記送信手段は、上記リーフエントリのそれぞれを識別する識別情報を上記第 2 の差分情報に付加して送信することを特徴とする送信装置。

【請求項 4】 コンテンツの所在を階層的に管理するディレクトリの階層構造を送信する送信方法において、

自分の配下の情報を格納可能なコンテナエントリと、上記コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理する管理のステップと、

上記管理のステップで管理される上記ディレクトリの階層構造の変化を検出し、該検出の結果に基づき、上記コンテナエントリの変化の差分からなる第 1 の差分情報と、上記リーフエントリの変化の差分からなる第 2 の差分情報とをそれぞれ

れ求める検出のステップと、

上記第 1 の差分情報と上記第 2 の差分情報とを分離して送信する送信のステップと

を有することを特徴とする送信方法。

【請求項 5】 送信された、コンテンツの所在を階層的に管理するディレクトリの階層構造を受信する受信装置において、

送信された、自分の配下の情報を格納可能なコンテナエントリと、上記コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造の、上記コンテナエントリの変化を検出して求められた第 1 の差分情報および該第 1 の差分情報に付加された上記コンテナエントリのそれぞれを識別する第 1 の識別情報と、上記リーフエントリの変化を検出して求められた第 2 の差分情報および該第 2 の差分情報に付加されたリーフエントリのそれぞれを識別する第 2 の識別情報とをそれぞれ受信する受信手段と、

上記第 1 の差分情報と上記第 2 の差分情報とに基づき構成されるディレクトリの階層構造を管理する管理手段と、

上記第 2 の差分情報を選択的に取り込み、取り込まれた該第 2 の差分情報に基づき、上記管理手段で管理される上記ディレクトリの階層構造を変更する変更手段と

を有することを特徴とする受信装置。

【請求項 6】 請求項 5 に記載の受信装置において、

上記変更手段は、上記第 1 の識別情報に対して選択的に設定された選択情報に基づき、該選択情報に対応した該第 1 の識別情報によって示される上記コンテナエントリの配下の上記リーフエントリによる上記第 2 の差分情報を選択的に取り込み、取り込まれた該第 2 の差分情報に基づき、上記管理手段で管理される上記ディレクトリの階層構造を変更するようにしたことを特徴とする受信装置。

【請求項 7】 送信された、コンテンツの所在を階層的に管理するディレクトリの階層構造を受信する受信方法において、

送信された、自分の配下の情報を格納可能なコンテナエントリと、上記コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリと

からなるディレクトリの階層構造の、上記コンテナエントリの変化を検出して求められた第 1 の差分情報および該第 1 の差分情報に付加された上記コンテナエントリのそれぞれを識別する第 1 の識別情報と、上記リーフエントリの変化を検出して求められた第 2 の差分情報および該第 2 の差分情報に付加されたリーフエントリのそれぞれを識別する第 2 の識別情報とをそれぞれ受信する受信のステップと、

上記第 1 の差分情報と上記第 2 の差分情報とに基づき構成されるディレクトリの階層構造を管理する管理のステップと、

上記第 2 の差分情報を選択的に取り込み、取り込まれた該第 2 の差分情報に基づき、上記管理のステップで管理される上記ディレクトリの階層構造を変更する変更のステップと

を有することを特徴とする受信方法。

【請求項 8】 コンテンツの所在を階層的に管理するディレクトリの階層構造を送信し、送信されたディレクトリの階層構造を受信する送受信システムにおいて、

自分の配下の情報を格納可能なコンテナエントリと、上記コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理する第 1 の管理手段と、

上記管理手段に管理される上記ディレクトリの階層構造の変化を検出し、該検出の結果に基づき、上記コンテナエントリの変化の差分からなる第 1 の差分情報と、上記リーフエントリの変化の差分からなる第 2 の差分情報とをそれぞれ求める検出手段と、

上記第 1 の差分情報に上記コンテナエントリのそれぞれを識別する第 1 の識別情報を付加して送信すると共に、該送信と分離して、上記第 2 の差分情報に上記リーフエントリのそれぞれを識別する第 2 の識別情報を付加して送信する送信手段と、

上記送信手段によって送信された上記第 1 の差分情報および上記第 1 の識別情報と上記第 2 の差分情報および上記第 2 の識別情報とをそれぞれ受信する受信手段と、

上記第1の差分情報と上記第2の差分情報とに基づき構成されるディレクトリの階層構造を管理する第2の管理手段と、

上記第2の差分情報を選択的に取り込み、取り込まれた該第2の差分情報に基づき、上記第2の管理手段で管理される上記ディレクトリの階層構造を変更する変更手段と

を有することを特徴とする送受信システム。

【請求項9】 請求項8に記載の送受信システムにおいて、

上記変更手段は、上記第1の識別情報に対して選択的に設定された選択情報に基づき、該選択情報に対応した該第1の識別情報によって示される上記コンテナエントリの配下の上記リーフエントリによる上記第2の差分情報を選択的に取り込み、取り込まれた該第2の差分情報に基づき、上記第2の管理手段で管理される上記ディレクトリの階層構造を変更するようにしたことを特徴とする送受信システム。

【請求項10】 コンテンツの所在を階層的に管理するディレクトリの階層構造を送信し、送信されたディレクトリの階層構造を受信する送受信方法において、

自分の配下の情報を格納可能なコンテナエントリと、上記コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理する第1の管理のステップと、

上記管理のステップによって管理される上記ディレクトリの階層構造の変化を検出し、該検出の結果に基づき、上記コンテナエントリの変化の差分からなる第1の差分情報と、上記リーフエントリの変化の差分からなる第2の差分情報とをそれぞれ求める検出のステップと、

上記第1の差分情報に上記コンテナエントリのそれぞれを識別する第1の識別情報を付加して送信すると共に、該送信と分離して、上記第2の差分情報に上記リーフエントリのそれぞれを識別する第2の識別情報を付加して送信する送信のステップと、

上記第1の送信のステップによって送信された上記第1の差分情報および上記第1の識別情報と、上記第2の送信のステップによって送信された上記第2の差

分情報および上記第 2 の識別情報とをそれぞれ受信する受信のステップと、

上記第 1 の差分情報と上記第 2 の差分情報とに基づき構成されるディレクトリの階層構造を管理する第 2 の管理のステップと、

上記第 1 の識別情報に対して選択的に設定された選択情報に基づき、該選択情報に対応した該第 1 の識別情報によって示される上記コンテナエントリの配下の上記リーフエントリによる上記第 2 の差分情報を選択的に取り込み、取り込まれた該第 2 の差分情報に基づき、上記第 2 の管理のステップによって管理される上記ディレクトリの階層構造を変更する変更のステップとを有することを特徴とする送受信方法。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

この発明は、階層構造を有すると共に、ネットワーク上に分散されて配置されたデータを一方向的に配信する際に用いて好適な送信装置および送信方法、受信装置および受信方法、ならびに、送受信システムおよび送受信方法に関する。

##### 【0002】

#### 【従来の技術】

データの配信方法としては、種々の方法が提案されており、例えば、現在のインターネット上においては、http (Hyper Text Transfer Protocol) のような、TCP/IP (Transmission Control Protocol/Internet Protocol) を基本とするプロトコルが採用されている。TCP/IP では、データの配信を受ける受信側からデータのデータの送信側に対して発呼が行われ、さらに、データの送受信を行う毎に、送信側と受信側との間でコネクションが確立されるようになっている。そのため、信頼性の高いデータの配信を行うことができる。

##### 【0003】

その反面で、送信側やネットワークの負荷が大きくなり、効率的なデータ配信を行うことが困難になる場合があった。すなわち、データの提供を受ける端末が増加し、データを提供するサーバへのアクセスが集中すると、サーバやネットワークに多大な負荷がかかり、データを要求しても、そのデータを得るまでに多大



な時間を要することがあった。

【0004】

そこで、データの配信を、例えば、広い地域にわたって一斉同報が可能な衛星回線やCATV(Cable Television)回線、実用化が予定されている地上波デジタル放送などを用いて行う方法が提案されている。この場合、端末の増加によって、サーバやネットワークに対する負荷が影響を受けることがない。

【0005】

一方、近年では、インターネットなどデジタル通信ネットワークの発達により、ネットワーク上に膨大なデータが蓄積されるようになり、この膨大なデータを効率的に利用することが求められている。そこで、ネットワーク上に分散されて存在するデータを階層的に管理し、ユーザに提供する、ディレクトリサービスが注目を集めている。ディレクトリサービスを利用することで、ユーザは、ネットワーク上に分散して存在する情報の中から、自分が必要な情報を素早く見つけ出し、アクセスすることが可能になる。

【0006】

ディレクトリサービスは、例えば、国際標準であるOSI(Open Systems Interconnection)などによって、X.500シリーズとして規定されている。X.500によれば、ディレクトリについて、開放型システムの集合体であり、各開放型システムが協調して、現実世界のオブジェクトの集合に関する情報の、論理的なデータベースを持つと定義されている。

【0007】

X.500によるディレクトリサービスによれば、ディレクトリが擁する情報の検索や閲覧を行うことができる。また、例えば電話帳に例えられる一覧やユーザの認証などもディレクトリサービスで提供される。さらに、ディレクトリサービスでは、特に人間のユーザにとって覚え易く、推測ならびに認識し易いように、オブジェクトの名前が付けられる。

【0008】

なお、このX.500によるディレクトリサービスは、極めて包括的なものであって、プログラムサイズが非常に大きく、TCP/IPをプロトコルとするイ

ンターネットでは実現が非常に難しい。そのため、LDAP (Lightweight Directory Access Protocol) といった、TCP/IP 向けに軽量化されたディレクトリサービスが提案されている。

#### 【0009】

ディレクトリサービスをユーザが利用する場合、ディレクトリに対してフィルタ処理を行うことができる。フィルタ処理を行うフィルタリングマスクは、ユーザの、ディレクトリへのアクセスの傾向や嗜好に応じて設定される。例えば、ユーザの嗜好に応じて特定の情報ジャンルに対してフィルタリングマスクが設定される。ユーザは、フィルタリングマスクが設定されたディレクトリ情報を、選択的にアクセスすることができる。フィルタ処理を行うことにより、ユーザが不要な情報を保持する必要が無くなる。

#### 【0010】

##### 【発明が解決しようとする課題】

ところで、近年では、上述した一斉同報を行うデータ伝送手段、すなわち、衛星回線やCATV回線、地上波デジタル放送などで、このディレクトリサービスを行うことが提案されている。この場合には、ディレクトリサービスによる情報が一方的に提供され、ユーザ側からの情報の要求ができない。したがって、ディレクトリサービスによるディレクトリ情報は、同一の情報が繰り返し送信される。

#### 【0011】

ユーザ側では、送信されてきた情報を、例えばテレビジョン受像機などに接続して用いられる、デジタル放送用受信機であるIRD (Integrated Receiver Decoder) や、STB (Set Top Box) に蓄積する。このとき、上述したフィルタリングマスクを用いてフィルタ処理がなされ、ユーザの嗜好に応じたディレクトリ情報が選択的に蓄積される。フィルタ処理に用いられるフィルタリングマスクは、ユーザ側で設定される。

#### 【0012】

ディレクトリの階層構造に変化が生じた場合には、ディレクトリサーバ側で実体的なディレクトリの更新を検知し、検知された階層構造の更新情報をユーザに

対して送信する。このとき、ユーザ側で、更新が検知された全階層構造の更新情報を処理するようにすると、ユーザ側での負担が大きくなる。特に、上述したSTBやIRDは、利用できる処理能力や記憶容量といった、リソースに乏しいため、STBやIRDを受信装置として利用する場合には問題になる。

【0013】

そこで、ユーザ側で、上述したフィルタリングマスクが設定された階層構造に関する更新情報だけを行うことで、ユーザ側での処理の負担を軽くすることが可能になる。しかしながら、この方法では、ディレクトリ構造そのものが動的に変化したような場合に、フィルタリングマスクの設定が意味をなさなくなってしまう、ユーザが必要としている情報を得ることができなくなってしまうおそれがあるという問題点があった。

【0014】

したがって、この発明の目的は、ユーザ側の負担が少なく、且つ、ディレクトリの階層構造の動的な変化に対応できるような送信装置および送信方法、受信装置および受信方法、ならびに、送受信システムおよび送受信方法を提供することにある。

【0015】

【課題を解決するための手段】

この発明は、上述した課題を解決するために、コンテンツの所在を階層的に管理するディレクトリの階層構造を送信する送信装置において、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあって、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理する管理手段と、管理手段に管理されるディレクトリの階層構造の変化を検出し、検出の結果に基づき、コンテナエントリの変化の差分からなる第1の差分情報と、リーフエントリの変化の差分からなる第2の差分情報とをそれぞれ求める検出手段と、第1の差分情報と第2の差分情報とを分離して送信する送信手段とを有することを特徴とする送信装置である。

【0016】

また、この発明は、コンテンツの所在を階層的に管理するディレクトリの階層

構造を送信する送信方法において、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理する管理のステップと、管理のステップで管理されるディレクトリの階層構造の変化を検出し、検出の結果に基づき、コンテナエントリの変化の差分からなる第 1 の差分情報と、リーフエントリの変化の差分からなる第 2 の差分情報とをそれぞれ求める検出のステップと、第 1 の差分情報と第 2 の差分情報とを分離して送信する送信のステップとを有することを特徴とする送信方法である。

## 【 0 0 1 7 】

また、この発明は、送信された、コンテンツの所在を階層的に管理するディレクトリの階層構造を受信する受信装置において、送信された、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造の、コンテナエントリの変化を検出して求められた第 1 の差分情報および第 1 の差分情報に付加されたコンテナエントリのそれぞれを識別する第 1 の識別情報と、リーフエントリの変化を検出して求められた第 2 の差分情報および第 2 の差分情報に付加されたリーフエントリのそれぞれを識別する第 2 の識別情報とをそれぞれ受信する受信手段と、第 1 の差分情報と第 2 の差分情報とに基づき構成されるディレクトリの階層構造を管理する管理手段と、第 2 の差分情報を選択的に取り込み、取り込まれた第 2 の差分情報に基づき、管理手段で管理されるディレクトリの階層構造を変更する変更手段とを有することを特徴とする受信装置である。

## 【 0 0 1 8 】

また、この発明は、送信された、コンテンツの所在を階層的に管理するディレクトリの階層構造を受信する受信方法において、送信された、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造の、コンテナエントリの変化を検出して求められた第 1 の差分情報および第 1 の差分情報に付加されたコンテナエントリのそれぞれを識別する第 1 の識別情報と、リーフエントリの変化を検出して求められた第 2 の差分情報および第 2 の差分情報

に付加されたリーフエントリのそれぞれを識別する第 2 の識別情報とをそれぞれ受信する受信のステップと、第 1 の差分情報と第 2 の差分情報とに基づき構成されるディレクトリの階層構造を管理する管理のステップと、第 2 の差分情報を選択的に取り込み、取り込まれた第 2 の差分情報に基づき、管理のステップで管理されるディレクトリの階層構造を変更する変更のステップとを有することを特徴とする受信方法である。

## 【 0 0 1 9 】

また、この発明は、コンテンツの所在を階層的に管理するディレクトリの階層構造を送信し、送信されたディレクトリの階層構造を受信する送受信システムにおいて、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理する第 1 の管理手段と、管理手段に管理されるディレクトリの階層構造の変化を検出し、検出の結果に基づき、コンテナエントリの変化の差分からなる第 1 の差分情報と、リーフエントリの変化の差分からなる第 2 の差分情報とをそれぞれ求める検出手段と、第 1 の差分情報にコンテナエントリのそれぞれを識別する第 1 の識別情報を付加して送信すると共に、該送信と分離して、第 2 の差分情報にリーフエントリのそれぞれを識別する第 2 の識別情報を付加して送信する送信手段と、第 1 の送信手段によって送信された第 1 の差分情報および第 1 の識別情報と、第 2 の送信手段によって送信された第 2 の差分情報および第 2 の識別情報とをそれぞれ受信する受信手段と、第 1 の差分情報と第 2 の差分情報とに基づき構成されるディレクトリの階層構造を管理する第 2 の管理手段と、第 2 の差分情報を選択的に取り込み、取り込まれた第 2 の差分情報に基づき、第 2 の管理手段で管理されるディレクトリの階層構造を変更する変更手段とを有することを特徴とする送受信システムである。

## 【 0 0 2 0 】

また、この発明は、コンテンツの所在を階層的に管理するディレクトリの階層構造を送信し、送信されたディレクトリの階層構造を受信する送受信方法において、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレ

クトリの階層構造を管理する第 1 の管理のステップと、管理のステップによって管理されるディレクトリの階層構造の変化を検出し、検出の結果に基づき、コンテナエントリの変化の差分からなる第 1 の差分情報と、リーフエントリの変化の差分からなる第 2 の差分情報とをそれぞれ求める検出のステップと、第 1 の差分情報にコンテナエントリのそれぞれを識別する第 1 の識別情報を付加して送信すると共に、該送信と分離して、第 2 の差分情報にリーフエントリのそれぞれを識別する第 2 の識別情報を付加して送信する送信のステップと、第 1 の送信のステップによって送信された第 1 の差分情報および第 1 の識別情報と、第 2 の送信のステップによって送信された第 2 の差分情報および第 2 の識別情報とをそれぞれ受信する受信のステップと、第 1 の差分情報と第 2 の差分情報とに基づき構成されるディレクトリの階層構造を管理する第 2 の管理のステップと、第 2 の差分情報を選択的に取り込み、取り込まれた第 2 の差分情報に基づき、第 2 の管理のステップによって管理されるディレクトリの階層構造を変更する変更のステップとを有することを特徴とする送受信方法である。

## 【 0 0 2 1 】

上述したように、請求項 1 および請求項 4 に記載の発明は、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理し、この階層構造の変化を検出した結果に基づき求められた、コンテナエントリの変化の差分からなる第 1 の差分情報とリーフエントリの変化の差分からなる第 2 の差分情報とを分離して送信するようにされているため、受信側において、コンテナエントリの差分情報とリーフエントリの差分情報とを、分離して処理することができる。

## 【 0 0 2 2 】

また、請求項 5 および請求項 6 に記載の発明は、送信された、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造の、コンテナエントリの変化を検出して求められた第 1 の差分情報および第 1 の差分情報に付加されたコンテナエントリのそれぞれを識別する第 1 の識別情報と、

リーフエントリの変化を検出して求められた第 2 の差分情報および第 2 の差分情報に付加されたリーフエントリのそれぞれを識別する第 2 の識別情報とをそれぞれ受信し、第 2 の差分情報を選択的に取り込み、取り込まれた第 2 の差分情報に基づき、第 1 および第 2 の差分情報とに基づき構成されるディレクトリ階層構造を変更するようにされているため、選択的に取り込まれた第 2 の差分情報だけを保持していればよく、余分な情報を保持する必要がない。

### 【0023】

また、請求項 7 および請求項 8 に記載の発明は、自分の配下の情報を格納可能なコンテナエントリと、コンテナエントリの配下にあつて、自分の配下の情報を格納できないリーフエントリとからなるディレクトリの階層構造を管理し、この階層構造の変化を検出した結果に基づき求められた、コンテナエントリの変化の差分からなる第 1 の差分情報および第 1 の差分情報に付加されたコンテナエントリのそれぞれを識別する第 1 の識別情報を送信すると共に、この送信と分離して、リーフエントリの変化の差分からなる第 2 の差分情報および第 2 の差分情報に付加されたリーフエントリのそれぞれを識別する第 2 の識別情報を付加して送信し、送信された第 1 の差分情報および第 1 の識別情報と、第 2 の差分情報および第 2 の識別情報とをそれぞれ受信し、第 2 の差分情報を選択的に取り込み、取り込まれた第 2 の差分情報に基づき、第 1 および第 2 の差分情報とに基づき構成されるディレクトリ階層構造を変更するようにされているため、受信側において、コンテナエントリの差分情報とリーフエントリの差分情報とを、分離して処理することができると共に、受信側は、選択的に取り込まれた第 2 の差分情報だけを保持していればよく、余分な情報を保持する必要がない。

### 【0024】

#### 【発明の実施の形態】

以下、この発明の実施の一形態を、図面を参照しながら説明する。図 1 は、この発明に適用できる系の一例を示す。送信側 1 は、例えばインターネットや放送ネットワークなどの、図示されないネットワーク上に存在する多数のコンテンツをツリー状の階層構造に整理し、ディレクトリ構造として管理している。送信側 1 では、このディレクトリ構造を示すディレクトリ情報を放送ネットワーク 2 に

対して送信する。受信側 3 は、図 2 に一例が示されるように、放送ネットワーク 2 に対して多数が接続され、放送ネットワーク 2 を介してなされた放送をそれぞれが受信可能とされている。受信側 3 は、後述する受信側ディレクトリサービスレプリケータ 17 によって放送ネットワーク 2 で放送されたディレクトリ情報を受信し、受信されたディレクトリ情報を参照して、放送ネットワーク 2 や他のネットワーク上に存在する多数の情報の中から必要な情報を選択し、入手することができる。

## 【0025】

図 1 に一例が示されるように、送信側 1 は、送信側ディレクトリサービスクライアント 10（以下、送信側クライアント 10 と略す）、送信側ディレクトリサーバ 11（以下、送信側サーバ 11 と略す）および送信側ディレクトリサーバレプリケータ 12（以下、送信側レプリケータ 12 と略す）からなる。これら送信側クライアント 10、送信側サーバ 11 および送信側レプリケータ 12 は、互いに例えばインターネットといったネットワークなどで接続されており、相互に通信が行われる。

## 【0026】

送信側クライアント 10 は、例えば図示されないネットワークなどによってコンテンツを提供するコンテンツプロバイダであって、ディレクトリ構造の変更や更新を行う。送信側クライアント 10 は、ネットワーク上の何処に位置していてもよい。送信側サーバ 11 は、送信側クライアント 10 の内容照会や変更などを行い、ディレクトリ構造を管理する。送信側サーバ 11 は、ネットワーク上で分散して構成することができる。送信側レプリケータ 12 は、送信側サーバ 11 で管理されているディレクトリ構造をモニタしてディレクトリ構造の更新を検出する。そして、送信側レプリケータ 12 は、この検出結果に基づき、更新前の構造と更新後の構造とを比較して差分を抽出し、ディレクトリ構造の差分更新情報を構成する。差分更新情報は、放送ネットワーク 2 に送信される。差分更新情報の構成については、後述する。

## 【0027】

受信側 3 は、受信側ディレクトリサーバレプリケータ 17（以下、受信側レプ



リケータ 17 と略す)、受信側ディレクトリサーバ 16 (以下、受信側サーバ 16 と略す) および受信側ディレクトリサービスクライアント 15 (以下、受信側クライアント 15 と略す) からなる。受信側 3 は、例えばパーソナルコンピュータや従来技術で述べた STB、IRD などであり、受信側クライアント 15 は、ディレクトリ構造にアクセスして複数の異なる形式のデータの取得ならびに表示ができるようにされた、例えば WWW(World Wide Web) ブラウザなどのアプリケーションソフトウェアである。また、受信側サーバ 16 は、ローカルなデータベースからなり、ディレクトリ情報が格納される。

#### 【0028】

放送ネットワーク 2 で送信されたディレクトリ情報、ディレクトリ構造の更新情報および更新情報の差分情報などは、受信側レプリケータ 17 に受信される。受信側レプリケータ 17 では、受信されたこれらの情報に基づき、受信側サーバ 16 に格納されたデータベースを更新し、ディレクトリ構造の再構築を行う。受信側クライアント 15 は、例えばユーザの指示により、受信側レプリケータ 17 に対して必要な情報を要求する。受信側レプリケータ 17 は、この要求に基づき受信側サーバ 16 のデータベースを検索して、受信側クライアント 15 に対して、例えば要求された情報のアドレスを返す。受信側クライアント 15 は、このアドレスに基づき、例えば図示されないネットワーク上に存在する情報にアクセスすることができる。

#### 【0029】

図 3 を用いて、ディレクトリ構造について説明する。ディレクトリは、図に示されるように、ツリー状の階層構造からなる。ツリーの各節点(ノード)をエントリと称し、各エントリには、情報が格納される。エントリには、ルートエントリ、コンテナエントリおよびリーフエントリの 3 種類が定義される。コンテナエントリは、さらに配下のエントリを包含することができるエントリである。コンテナエントリによって構成される階層を、以下、コンテナ階層と称する。

#### 【0030】

コンテナエントリ以外のエントリを、リーフエントリと称する。リーフエントリは、配下にエントリを含むことができない、末端の節点である。リーフエン

リによる階層を、以下、リーフ階層と称する。リーフ階層は、コンテナエントリの配下に構成される。

#### 【 0 0 3 1 】

また、ディレクトリツリーの最上位のエントリは、ルートエントリと称され、当該ディレクトリ構造で完結される世界全体を示すエントリである。なお、以下では、コンテナエントリは、少なくとも一つのリーフエントリあるいはコンテナエントリを配下に持つものとする。

#### 【 0 0 3 2 】

エントリは、複数の属性を持つ。エントリが持つ属性のうちで、ディレクトリツリーで一意に識別される名前を、エントリ名と称する。エントリ名によって、そのエントリの、ディレクトリ構造上の位置を特定することができる。図 3 の例では、ルートエントリには、エントリ名 A が与えられている。ルートエントリの直接的な配下のエントリには、図の左側に示されるリーフエントリにはエントリ名 A、B が、右側のコンテナエントリにはエントリ名 A、C がそれぞれ与えられる。以下、ルートエントリから階層構造を辿った順にピリオドで区切られて、各エントリに対してエントリ名が与えられる。

#### 【 0 0 3 3 】

図 4 は、コンテナエントリの構造の一例を示す。コンテナエントリは、コンテナエントリ自身の属性と、自分の配下のコンテナエントリおよびリーフエントリのリストとを有する。配下のエントリのリストは、要素を含まないこともできる。また、コンテナエントリ自身の属性は、図示されるように、複数持つことができる。

#### 【 0 0 3 4 】

図 5 は、リーフエントリの構造の一例を示す。リーフエントリは、図 5 A に示されるように、リーフエントリの属性を複数、有する。図 5 B は、リーフエントリの属性のより具体的な例である。各属性は、属性名と属性値とからなる。例えばリーフエントリがコンテンツの検索情報である場合には、属性名の一つにアドレスがあり、その属性値として URL (Uniform Resource Locator) などのコンテンツのアドレス情報が記述される。

## 【 0 0 3 5 】

ディレクトリ構造は、例えば、そのディレクトリ構造で完結する世界全体を表すルートエントリの下に、コンテナエントリが例えば情報ジャンルに応じてツリー状に分類され配されて、構成される。

## 【 0 0 3 6 】

送信側 1 の構成について、より具体的に説明する。送信側サーバ 1 1 には、図 3 ～図 5 で既に説明したような構成に準えて、ディレクトリ構造が管理されている。送信側クライアント 1 0 は、提供するコンテンツに応じて、送信側サーバ 1 1 で管理されているディレクトリ構造に対して変更などを加える。送信側サーバ 1 1 に対してなされた変更は、送信側レプリケータ 1 2 にモニタされる。

## 【 0 0 3 7 】

図 6 は、送信側レプリケータ 1 2 の機能を説明するための機能ブロック図である。送信側レプリケータ 1 2 は、例えば一般的なコンピュータシステムで構成することができ、CPU (Central Processing Unit) 、メモリ、ハードディスクといった記録および記憶媒体、通信手段、タイマ、ユーザインターフェイスなどからなる。この図 6 に示される機能ブロックは、CPU 上で動作するアプリケーションソフトウェアにより実現され、各モジュールは、アプリケーションソフトウェア上の機能的な単位であって、それぞれがソフトウェアからなる。

## 【 0 0 3 8 】

送信側レプリケータ 1 2 は、更新検知モジュール 2 0、メッセージ生成モジュール 2 1 およびメッセージ放送モジュール 2 2 からなる。これらモジュール 2 0、2 1 および 2 2 のそれぞれは、コンテナ階層に関する処理を行うモジュールと、リーフ階層に関する処理を行うモジュールとを有する。

## 【 0 0 3 9 】

更新検知モジュール 2 0 は、送信側サーバ 1 1 を参照して、サーバ 1 1 上に管理されているディレクトリ構造に変化があったかどうかを検知するモジュールで、コンテナ階層更新検知モジュール 2 3 とリーフ階層更新検知モジュール 2 4 とからなる。コンテナ階層更新検知モジュール 2 3 は、送信側サーバ 1 1 をモニタして、コンテナ階層の構造の変化を検知する。また、リーフ階層更新検知モジュ

ール 2 4 は、送信側サーバ 1 1 をモニタして、リーフ階層の構造およびリーフエントリの内容の変化を検知する。

#### 【 0 0 4 0 】

メッセージ生成モジュール 2 1 は、更新検知モジュール 2 0 によるディレクトリ構造の変化の検知結果に基づく、ディレクトリ構造の差分更新情報が示されたメッセージを生成するモジュールで、コンテナストラクチャアップデートメッセージ生成モジュール 2 5 と、リーフエントリアップデートメッセージ生成モジュール 2 6 とからなる。コンテナストラクチャアップデートメッセージ生成モジュール 2 5 は、コンテナ階層更新検知モジュール 2 3 の検知結果に基づき、コンテナ階層における構造変化に関する差分更新情報が示されたメッセージを生成する。また、リーフエントリアップデートメッセージ生成モジュール 2 6 は、リーフ階層更新検知モジュール 2 4 の検知結果に基づき、リーフ階層における更新情報が示されたメッセージを生成する。

#### 【 0 0 4 1 】

メッセージ放送モジュール 2 2 は、メッセージ生成モジュール 2 1 で生成されたメッセージを放送ネットワーク 2 に対して放送するモジュールで、コンテナストラクチャアップデートメッセージ放送モジュール 2 7 とリーフエントリアップデートメッセージ放送モジュール 2 8 とからなる。コンテナストラクチャアップデートメッセージ放送モジュール 2 7 は、上述した、コンテナストラクチャアップデートメッセージ生成モジュール 2 5 で生成されたメッセージを放送する。また、リーフエントリアップデートメッセージ放送モジュール 2 8 は、上述した、リーフエントリアップデートメッセージ生成モジュール 2 6 で生成されたメッセージを放送する。なお、メッセージ放送モジュール 2 2 からの放送ネットワーク 2 に対するメッセージの放送は、同一のメッセージが所定回数だけ、サイクリックに送信されて行われる。

#### 【 0 0 4 2 】

次に、受信側 3 の構成について、より具体的に説明する。図 7 は、受信側クライアント 1 5 の機能を説明するための機能ブロック図である。受信側クライアント 1 5 は、例えば一般的なコンピュータシステムで構成することができ、CPU

(Central Processing Unit)、メモリ、ハードディスクといった記録および記憶媒体、通信手段、ユーザインターフェイスなどからなる。この図 7 に示される機能ブロックは、CPU 上で動作するアプリケーションソフトウェアにより実現され、各モジュールは、アプリケーションソフトウェア上の機能的な単位であって、それぞれがソフトウェアからなる。

#### 【0043】

受信側クライアント 15 は、上述したように、例えば WWW ブラウザであり、供給されたコンテンツ、例えば画像データ、テキストデータ、音声データおよび動画データを統一的に表示および再生することができるようにされている。また、所定の入力手段を用いてユーザによって入力された指示に基づき、上述の表示ならびに再生を制御することができる。

#### 【0044】

受信側クライアント 15 は、ディレクトリ検索モジュール 30、ユーザ対話管理モジュール 31 およびコンテンツ取得モジュール 32 とからなる。また、ユーザ対話管理モジュール 31 に対して、例えばキーボードなどのテキスト入力手段、マウスなどのポインティングデバイスや表示装置などからなるユーザインターフェイス 33 が接続される。ユーザの、受信側クライアント 15 に対するコンテンツ検索要求の入力は、ユーザインターフェイス 33 を用いて、ユーザ対話モジュール 31 に対して対話形式で行われる。

#### 【0045】

ユーザ対話モジュール 31 に対してコンテンツ検索要求が入力されると、ユーザ対話管理モジュール 31 からディレクトリ検索モジュール 30 に対して、コンテンツのアドレスを検索するため、コンテンツに対応するディレクトリエントリを検索するよう依頼が出される。ディレクトリ検索モジュール 30 では、この検索依頼に応じて、受信側サーバ 16 に対してディレクトリエントリ検索要求を送る。

#### 【0046】

受信側サーバ 16 での検索要求に基づくディレクトリエントリの検索結果がディレクトリ検索モジュール 30 に返される。検索結果は、ディレクトリ検索モジ

ユーザ 30 からユーザ対話管理モジュール 31 に返される。そして、検索結果のディレクトリエントリ情報から、例えば検索結果がリーフエントリであれば、属性の一つであるコンテンツのアドレス情報が取り出される。ユーザ対話管理モジュール 31 からコンテンツ取得モジュール 32 に対して、取り出されたアドレス情報によって示されるコンテンツを取得するよう、コンテンツ取得依頼が出される。

## 【0047】

コンテンツ取得モジュール 32 では、受け取ったコンテンツ取得依頼に基づき、コンテンツサーバ 35 に対してコンテンツの取得要求を送る。コンテンツサーバ 35 は、受信側クライアント 15 と、例えばインターネットといった双方向ネットワーク 36 を介して接続されるサーバであり、ユーザに対してコンテンツを提供する。コンテンツの提供は、双方向ネットワーク 36 を介して行ってもいいし、放送ネットワーク 2 によって行うこともできる。

## 【0048】

コンテンツ取得要求に基づきコンテンツサーバ 35 から取得されたコンテンツは、例えば双方向ネットワーク 36 を介してコンテンツ取得モジュール 32 に供給され、コンテンツ取得モジュール 32 からユーザ対話管理モジュール 31 に返される。ユーザ対話管理モジュール 31 では、受け取ったコンテンツをユーザインターフェイス 33 に出力する。

## 【0049】

なお、要求するコンテンツが放送ネットワーク 2 で伝送されるものである場合には、コンテンツ取得モジュール 32 は、放送ネットワーク 2 で放送される所望のコンテンツを、コンテンツ取得依頼に基づき直接的に取得するようにしてもよい。

## 【0050】

図 8 は、受信側サーバ 16 の機能を説明するための機能ブロック図である。この受信側サーバ 16 も、説明は省略するが、上述の受信側クライアント 15 と同様に、一般的なコンピュータシステムによって構成される。受信側サーバ 16 は、ディレクトリ更新要求処理モジュール 40、ディレクトリデータベース 41 お

よびディレクトリ検索要求処理モジュール42からなる。

【0051】

ディレクトリデータベース41は、送信側サーバ11で管理されるディレクトリ構造に基づくディレクトリ情報が格納されている。上述したように、受信側レプリケータ17は、放送ネットワーク2を介して送信側1から送信されたディレクトリ構造の差分更新情報を受信する。詳細は後述するが、受信側レプリケータ17からディレクトリ更新要求処理モジュール40に対して、差分更新情報に基づきディレクトリデータベース41に格納されているディレクトリ情報の更新を行うよう、要求が出される。ディレクトリ更新要求処理モジュール40では、この要求に基づき、差分更新情報を用いてディレクトリデータベース41に格納されているディレクトリ情報の更新を行う。

【0052】

一方、上述した受信側クライアント15からのディレクトリエントリの検索要求は、ディレクトリ検索要求処理モジュール40に受け取られる。ディレクトリ検索要求処理モジュール40によって、受け取った検索要求に基づきディレクトリデータベース41が検索される。検索した結果得られたディレクトリエントリ、例えばリーフエントリ中のアドレス情報は、ディレクトリ検索要求処理モジュール42から受信側クライアント15に返される。

【0053】

上述したように系を構成することで、ユーザは、受信側クライアント15によってディレクトリ情報を検索し、検索結果として、所望のコンテンツが提供されるアドレス情報を得ることができる。そして、ユーザは、得られたアドレス情報に基づき所望のコンテンツを取得することができる。また、ディレクトリ構造は、常に送信側レプリケータ12によってモニタされ、ディレクトリ構造に変更が生じたときには、変更に伴うディレクトリ構造の差分更新情報が放送ネットワーク2を介して受信側レプリケータ17に供給される。ユーザ側では、供給された差分更新情報に基づき、受信側レプリケータ17によってディレクトリデータベース41に格納されたディレクトリ情報が変更される。そのため、ユーザは、常に実際のディレクトリ構造と同期したディレクトリ情報を、ディレクトリデータ

ベース 41 に保持することができる。

【0054】

次に、図 9 および図 10 を用いて、ディレクトリ構造の変化に伴い生成される、ディレクトリ構造の差分更新情報について説明する。以下では、スキーマバージョン  $S_v$  で特定されるあるコンテナ階層のコンテナエントリ  $X$  の配下に、コンテナエントリ  $C$  またはリーフエントリ  $l$  を追加または削除する処理を、

( $S_v, X, [+/-] [C/l]$ )

と記述する。このディレクトリ構造に対する処理を示す記述は、この記述による処理で変化したディレクトリ構造の、元の構造との差分を表しており、これを差分更新情報として用いることができる。

【0055】

スキーマバージョン  $S_v$  は、ディレクトリ構造の変更に伴い変化する値である。コンテナエントリ  $X$  (または  $C$ ) は、コンテナエントリ名であり、ここでは大文字のアルファベットで表す。リーフエントリ  $l$  は、リーフエントリ名であり、ここでは小文字のアルファベットで表す。エントリの追加は  $[+]$  で表され、削除は  $[-]$  で表される。括弧  $[]$  中のスラッシュ記号は、その両側に記された何方かが記述されることを示す。なお、図 9 および図 10 において、二重線の四角は、コンテナエントリを表し、単線の四角は、リーフエントリを表す。ルートエントリは、接続線だけが示され、本体は省略されている。

【0056】

図 9 A において、図示されないルートエントリの配下にコンテナエントリ  $A$  のみが存在している。この状態をスキーマバージョン  $S_v = 1$  とする。この状態に、上述の記述に従い  $(1, A, +B)$  の処理を行う。すなわち、コンテナエントリ  $A$  の配下にコンテナエントリ  $B$  が加えられる。すると、図 9 B のようなディレクトリ構造になる。図 9 A の状態にコンテナエントリ  $B$  が加えられたことで、コンテナエントリの階層イメージが変化したとされ、スキーマバージョン  $S_v = 2$  とされる。

【0057】

図 9 B の状態に、さらに  $(2, A, +a)$  の処理を行う。すなわち、コンテナ



エントリ A の配下にリーフエントリ a を加える。すると、図 9 C のようなディレクトリ構造になる。さらにまた、図 9 C の状態に (2, A, -a) の処理を行う。すなわち、リーフエントリ a を、コンテナエントリ A の配下から削除する。すると、図 9 D のようなディレクトリ構造になる。さらに、図 9 D の状態に (2, A, -B) の処理を行う。すなわち、コンテナエントリ B を、コンテナエントリ A の配下から削除する。すると、図 9 E のようなディレクトリ構造になる。

## 【0058】

図 9 E の状態では、コンテナエントリの階層イメージが図 9 D の状態から変化しているため、スキーマバージョン S<sub>v</sub> が更新され、スキーマバージョン S<sub>v</sub> = 3 になる。したがって、図 9 E の状態において、コンテナエントリ A の配下にコンテナエントリ C を加える処理は、(3, A, +C) と記述できる。この処理を行うと、図 9 F に示されるディレクトリ構造となる。

## 【0059】

この図 9 の例では、(1, A, +B)、(2, A, +a)、(2, A, -a)、(2, A, -B) および (3, A+C) がそれぞれの段階での差分更新情報とされる。

## 【0060】

図 10 は、ディレクトリ構造を変更する別の例を示す。上述の図 9 に示される例では、一度に一つの処理を行っていたが、図 10 では、2 つずつの処理をまとめて行っている。図 10 A は、図示されないルートエントリの配下にコンテナエントリ A のみが存在している。この状態がスキーマバージョン S<sub>v</sub> = 1 とする。この図 10 A の状態に対して、(1, A, +B) および (1, A, +a) の処理を順に行う。すなわち、コンテナエントリ A の配下に、コンテナエントリ B とリーフエントリ a とが追加される。すると、図 10 B のようなディレクトリ構造となる。コンテナエントリの階層イメージが変わり、スキーマバージョン S<sub>v</sub> = 2 となる。

## 【0061】

図 10 B の状態に、さらに、(2, A, -a) および (2, B, +b) の処理を順に行う。すなわち、コンテナエントリ A の配下からリーフエントリ a を削除

し、その後、コンテナエントリ B の配下にリーフエントリ b を追加する。すると、図 10 C のようなディレクトリ構造となる。

#### 【0062】

図 10 C の状態に対して、さらに、(2, B, +C) および (2, C, +c) の処理を行う。すなわち、コンテナエントリ B の配下にコンテナエントリ C を追加した後に、追加されたコンテナエントリ C の配下にリーフエントリ c を追加する。この処理においては、追加されたコンテナエントリに対してさらにエントリが追加されるため、処理の前後を入れ換えることができない。処理後、図 10 D のようなディレクトリ構造となり、コンテナエントリの階層イメージが変わったため、スキーマバージョン S v が更新され、スキーマバージョン S v = 3 とされる。

#### 【0063】

この図 10 の例では、(1, A, +B) および (1, A, +a)、(2, A, -a) および (2, B, +b)、ならびに、(2, B, +C) および (2, C, +c) がそれぞれの段階での差分更新情報とされる。上述したように、複数の処理を一つのディレクトリ構造の更新処理としてまとめて行うときには、処理の順序を考慮する必要がある。

#### 【0064】

なお、ディレクトリ構造の差分更新情報は、上述の例に限定されず、適用されるシステムに応じて様々な形式とすることができる。

#### 【0065】

また、リーフエントリに関しては、コンテナエントリの配下からの削除やコンテナエントリの配下への追加の他に、内容の修正だけが行われることがある。内容の修正だけが行われた場合には、ディレクトリ構造における変化は、生じない。この場合には、例えば、リーフエントリ名と、当該リーフエントリ中で修正のあった属性名および属性値の列で、差分更新情報が構成される。一例として、

```
{
  LeafEntryName,
  Set of { AttributeName, AttributeValue }
```

}

このように差分更新情報が記述される。

## 【 0 0 6 6 】

この発明が適用される系では、上述したように、差分更新情報は、送信側 1 から受信側 3 に対して放送ネットワーク 2 を介して一方向的に送信される。また、一つの送信側 1 に対して複数の受信側 3 が存在し、複数の受信側 3 の稼働状態もそれぞれ異なる。そのため、送信側 1 で管理されているディレクトリ情報と、受信側 3 で管理されているディレクトリ情報とを同期させる必要がある。

## 【 0 0 6 7 】

以下、送信側 1 の送信側サーバ 1 1 に格納されたディレクトリ情報と、受信側 3 の受信側サーバ 1 6 に格納されたディレクトリ情報とを同期させ、ディレクトリ構造の同期管理方法について説明する。

## 【 0 0 6 8 】

最初に、図 1 1 のフローチャートを用いてコンテナエントリの同期管理方法について説明する。まず、ステップ S 1 で、送信側クライアント 1 0 によって、送信側サーバ 1 1 で管理されているディレクトリ構造の、コンテナ階層の構成が変更される。例えば、コンテナエントリの配下に新たなコンテナエントリやリーフエントリが追加される処理や、コンテナエントリの配下のコンテナエントリやリーフエントリが削除される処理が行われる。

## 【 0 0 6 9 】

次のステップ S 2 では、送信側サーバ 1 1 に対してなされた変更が送信側レプリケータ 1 2 によって検知され、検知結果に基づき、コンテナ階層構成の変更によるコンテナ構造更新情報 M s g . 1 が生成される。生成されたコンテナ構造更新情報 M s g . 1 は、放送ネットワーク 2 に対して放送される。コンテナ構造更新情報 M s g . 1 の放送は、同一の内容が所定回数、サイクリックに繰り返されて行われる。

## 【 0 0 7 0 】

放送されたコンテナ構造更新情報 M s g . 1 は、ステップ S 3 で、受信側レプリケータ 1 7 によって受信される。受信側レプリケータ 1 7 は、受信側サーバ 1

6に格納されたディレクトリ情報に管理されるコンテナ階層構成を、受信したコンテナ構造更新情報Msg. 1に基づき変更する。これにより、送信側1と受信側3とで、ディレクトリ情報のコンテナ階層の構造の同期がとられる。

#### 【0071】

コンテナ構造更新情報Msg. 1のフォーマットは、例えば、  
 Container Structure Update Message {  
     MessageID,  
     差分更新情報  
 }

このように定義される。「MessageID」(メッセージID)は、このメッセージ(コンテナ構造更新情報Msg. 1)の識別情報であって、例えば、このメッセージが生成される毎に1ずつ増加される整数である。また、「差分更新情報」は、コンテナ階層構成の変更による、上述したディレクトリ構造の差分更新情報である。

#### 【0072】

図11のフローチャートにおけるステップS2の処理を、図12のフローチャートを用いて、より詳細に説明する。この図12のフローチャートによる処理は、全て送信側レプリケータ12上で行われる。先ず、ステップS10で、送信側サーバ11上のコンテナエントリの階層構成の情報が全て読み込まれる。読み込まれたコンテナエントリの階層構成の情報は、送信側レプリケータ12が有する、例えばメモリやハードディスクといった記録または記憶媒体に、コピー1として記憶される。

#### 【0073】

コピー1が記憶されたら、次のステップS11で、タイマが所定の時間にセットされ、起動される。ステップS12によって、タイマにセットされた所定時間を超過したかどうか判断され、所定時間を超過したと判断されたなら、処理はステップS13に移行する。ステップS13では、再び送信側サーバ11上のコンテナエントリの階層構成の情報が全て読み込まれる。読み込まれたコンテナエントリの階層構成は、送信側レプリケータ12が有する、例えばメモリやハード

ディスクといった記録または記憶媒体に、コピー 2 として記憶される。

【0074】

次のステップ S 14 では、ステップ S 10 で記憶されたコピー 1 と、ステップ S 13 で記憶されたコピー 2 とが比較される。比較の結果、両者に差分が無いとされれば（ステップ S 15）、処理はステップ S 11 へ戻され、再びタイマがセットされ、コピー 2 の記憶がなされる。

【0075】

一方、ステップ S 14 で、コピー 1 とコピー 2 との間に差分があるとされれば、処理はステップ S 16 に移行する。ステップ S 16 では、コピー 1 とコピー 2 との差分に基づき、差分更新情報が生成される。そして、この差分更新情報が記述されたコンテナ構造更新情報 M s g. 1 が生成される。生成されたコンテナ構造更新情報 M s g. 1 は、放送ネットワーク 2 に対して送信され、放送される。放送されたコンテナ構造更新情報 M s g. 1 は、受信側レプリケータ 17 に受信される。

【0076】

ステップ S 16 でコンテナ構造更新情報 M s g. 1 が放送されると、次のステップ S 17 で、コピー 1 の内容がコピー 2 の内容で置き換えられ、処理はステップ S 11 に戻される。

【0077】

図 11 のフローチャートにおけるステップ S 3 の処理を、図 13 のフローチャートを用いて、より詳細に説明する。この図 13 のフローチャートによる処理は、全て受信側レプリケータ 17 上で行われる。最初のステップ S 20 で、送信側レプリケータ 12 によって放送ネットワーク 2 を介して放送されたコンテナ構造更新情報 M s g. 1 が、受信側レプリケータ 17 によって受信される。

【0078】

ステップ S 21 で、ステップ S 20 での受信がコンテナ構造更新情報 M s g. 1 の初回の受信かどうか判断される。そして、この受信が初回の受信では無いと判断されたら、処理はステップ S 23 に移行し、受信されたコンテナ構造更新情報 M s g. 1 のメッセージ ID が受信側レプリケータ 17 が有する、例えばメ

モリやハードディスクといった記録または記憶媒体に、コピー 3 として記憶される。

【 0 0 7 9 】

次のステップ S 2 4 では、受信されたコンテナ構造更新情報 M s g . 1 の内容、すなわち、コンテナ構造更新情報 M s g . 1 に記述された差分更新情報に基づき、受信側サーバ 1 6 で管理されているディレクトリ情報が更新され、そのディレクトリ情報で示されるコンテナ階層の構成が変更される。ステップ S 2 4 の処理の後、処理はステップ S 2 0 に戻される。

【 0 0 8 0 】

一方、上述のステップ S 2 1 で、ステップ S 2 0 でのコンテナ構造更新情報 M s g . 1 の受信が初回の受信では無いと判断されたら、処理はステップ S 2 2 に移行する。ステップ S 2 2 では、受信されたコンテナ構造更新情報 M s g . 1 に記述されるメッセージ I D と、前回の受信の際のステップ S 2 3 の処理でコピー 3 として記憶されたメッセージ I D とが同一であるかどうか判断される。若し、同一であるとされれば、処理はステップ S 2 0 に戻される。

【 0 0 8 1 】

一方、ステップ S 2 2 で、両者のメッセージ I D が同一では無いとされれば、処理はステップ S 2 3 に移行する。ステップ S 2 3 では、上述したように、メッセージ I D が記憶媒体にコピー 3 として記憶される。この場合には、新たに受信されたメッセージ I D で、前回受信され記憶されたメッセージ I D が例えば上書きされることになる。そして、次のステップ S 2 4 で、受信されたコンテナ構造更新情報 M s g . 1 に基づき、受信側サーバ 1 6 上のコンテナエントリ階層の内容が変更される。

【 0 0 8 2 】

次に、図 1 4 のフローチャートを用いてリーフエントリの同期管理方法について説明する。まず、ステップ S 3 0 で、送信側クライアント 1 0 によって、送信側サーバ 1 1 で管理されているディレクトリ構造の、あるコンテナエントリの配下のリーフエントリが変更される。例えば、あるコンテナエントリの配下の新たなリーフエントリの追加や、あるコンテナエントリの配下のリーフエントリの削

除や修正といった処理が行われる。

【0083】

次のステップS31では、送信側サーバ11のあるコンテナエントリの配下のリーフエントリに対してなされた変更が送信側レプリケータ12によって検知される。そして、検知結果に基づき、あるコンテナエントリ配下のリーフエントリの変更によるリーフ更新情報Msg. x1が生成される。生成されたリーフ更新情報Msg. x1は、放送ネットワーク2を介して複数の受信側レプリケータ17に対してサイクリックに放送される。

【0084】

放送されたリーフ更新情報Msg. x1は、ステップS32で、受信側レプリケータ17によって受信される。受信側レプリケータ17は、受信したリーフ更新情報Msg. x1に基づき、受信側サーバ16に格納されたディレクトリ情報に管理される、対応するリーフエントリを変更する。これにより、送信側1と受信側3とで、ディレクトリ情報のリーフエントリの同期がとられる。

【0085】

リーフ更新情報Msg. x1のフォーマットは、例えば、

```
Leaf Entry Update Message {
    MessageID,
    差分更新情報
}
```

このように定義される。「MessageID」（メッセージID）は、このメッセージ（リーフ更新情報Msg. x1）の識別情報であって、例えば、このメッセージが生成される毎に1ずつ増加される整数である。また、「差分更新情報」は、上述したディレクトリ構造の差分更新情報である。

【0086】

図14のフローチャートにおけるステップS31の処理を、図15のフローチャートを用いて、より詳細に説明する。この図15のフローチャートによる処理は、全て送信側レプリケータ12上で行われる。まず、ステップS40で、送信側サーバ11上の、あるコンテナエントリの配下のリーフエントリ名が全て読み

込まれる。読み込まれたリーフエントリ名は、送信側レプリケータ 12 が有する、例えばメモリやハードディスクといった記録または記憶媒体に、コピー 4 として記憶される。

## 【0087】

コピー 1 が記憶されたら、次のステップ S 4 1 で、タイマが所定の時間にセットされ、起動される。ステップ S 4 2 によって、タイマにセットされた所定時間を超過したかどうかが判断され、所定時間を超過したと判断されたなら、処理はステップ S 4 3 に移行する。ステップ S 4 3 では、再び送信側サーバ 11 上の、あるコンテナエントリの配下のリーフエントリ名が全て読み込まれる。読み込まれたリーフエントリ名は、送信側レプリケータ 12 が有する、例えばメモリやハードディスクといった記録または記憶媒体に、コピー 5 として記憶される。

## 【0088】

次のステップ S 4 4 では、ステップ S 4 0 で記憶されたコピー 4 と、ステップ S 4 3 で記憶されたコピー 5 とが比較される。比較の結果、両者の間に差分が無いとされれば（ステップ S 4 5）、処理はステップ S 4 1 へ戻され、再びタイマがセットされ、コピー 5 の記憶が行われる。

## 【0089】

一方、ステップ S 4 4 で、コピー 4 とコピー 5 との間に差分があるとされれば、処理はステップ S 4 6 に移行する。ステップ S 4 6 では、コピー 4 とコピー 5 との差分に基づき、差分更新情報が生成される。そして、この差分更新情報が記述されたリーフ更新情報 M s g. x 1 が生成される。生成されたリーフ更新情報 M s g. x 1 は、放送ネットワーク 2 に対して送信され、放送される。放送されたリーフ更新情報 M s g. x 1 は、複数の受信側レプリケータ 17 に受信される。

## 【0090】

ステップ S 4 6 でリーフ更新情報 M s g. x 1 が放送されると、次のステップ S 4 7 で、コピー 4 の内容がコピー 5 の内容で書き替えられ、処理はステップ S 4 1 に戻される。



## 【0091】

なお、上述の図15のフローチャートの処理は、送信側レプリケータ12によって、送信側サーバ11が管理するディレクトリ構造上の全てのコンテナエントリに対して行われる。

## 【0092】

図14のフローチャートにおけるステップS32の処理を、図16のフローチャートを用いて、より詳細に説明する。この図16のフローチャートによる処理は、全て受信側レプリケータ17上で行われる。最初のステップS50で、送信側レプリケータ12によって放送ネットワーク2を介して放送されたリーフ更新情報Msg. x1が、受信側レプリケータ17によって受信される。

## 【0093】

ステップS51で、ステップS50での受信がリーフ更新情報Msg. x1の初回の受信であるかどうか判断される。若し、この受信が初回の受信で無いと判断されたら、処理はステップS53に移行し、受信されたリーフ更新情報Msg. x1のメッセージIDが受信側レプリケータ17が有する、例えばメモリやハードディスクといった記録または記憶媒体に、コピー6として記憶される。

## 【0094】

次のステップS54では、受信されたリーフ更新情報Msg. x1の内容、すなわち、リーフ更新情報Msg. x1に記述された差分更新情報に基づき、受信側サーバ16で管理されているディレクトリ情報のうち、対応するリーフエントリの内容が変更される。ステップS54の処理の後、処理はステップS50に戻される。

## 【0095】

一方、上述のステップS51で、ステップS50でのリーフ更新情報Msg. x1の受信が初回の受信では無いと判断されたら、処理はステップS52に移行する。ステップS52では、受信されたリーフ更新情報Msg. x1に記述されるメッセージIDと、前回の受信の際のステップS53の処理でコピー6として記憶されたメッセージIDとが同一であるかどうか判断される。若し、同一であるとされれば、処理はステップS50に戻される。

## 【0096】

一方、ステップS52で、両者のメッセージIDが異なるとされれば、処理はステップS53に移行する。ステップS53では、上述したように、メッセージIDが記憶媒体にコピー6として記憶される。この場合には、新たに受信されたメッセージIDで、前回受信され記憶されたメッセージIDが例えば上書きされることになる。そして、次のステップS54で、受信されたリーフ更新情報Msg. x1に基づき、受信側サーバ16上の対応するリーフエントリの内容が変更される。

## 【0097】

ところで、上述したように、図14のフローチャート中のステップS31による、リーフ更新情報Msg. x1の放送は、送信側1で管理されるディレクトリ構造中の、全コンテナエントリのそれぞれに関して行われ、放送されるリーフ更新情報Msg. x1は、膨大な量になることが予想される。したがって、従来技術で問題点として既に述べたように、受信側3において全てのリーフ更新情報Msg. x1を受信し、図16のフローチャートによる処理を行うことは、大きな負担になる。そのため、受信側3では、必要とされている、すなわち、頻繁に照会されるコンテナエントリの配下のリーフエントリに対するリーフ更新情報Msg. x1のみを、放送された多数のリーフ更新情報Msg. x1から、効率よくフィルタ処理する必要がある。

## 【0098】

例えば、受信側レプリケータ17が家庭内でテレビジョン受像機などに接続して用いられる、セットトップボックス(STB)のような、処理能力や記憶容量が限定された、すなわち、コンピュータリソースに制約のある環境に実装される場合を想定する。この場合には、放送されるリーフ更新情報Msg. x1の取得には限度がある。そのため、受信されたリーフ更新情報Msg. x1を取捨選択して装置内に取り込み、記憶コストやメッセージ処理コストの軽減を図る必要がある。すなわち、受信側レプリケータ17において、不必要な記憶や処理にかかるコストを制限する必要がある。特に、ディレクトリサービスが普及し、送信側サーバ11で管理される対象のディレクトリ構造が巨大になるのに伴い、上述の

リーフ更新情報Msg. x 1の取捨選択は、より重要な事項となってくる。

【0099】

以下に、リーフ更新情報Msg. x 1に対してフィルタ処理を行う方法について説明し、さらに、この発明の主旨に係る、フィルタ処理を効率的に行う方法について説明する。送信側レプリケータ12は、放送されるリーフ更新情報Msg. x 1に対して、受信側レプリケータ17においてフィルタ処理を行うためのフィルタリングマスクを付加する。フィルタリングマスクを解釈するためのマスクスキーマ構造と、マスクスキーマ構造を送信側レプリケータ12から受信側レプリケータ17に通知する方法などについては、後述する。

【0100】

リーフ更新情報Msg. x 1にフィルタリングマスクを付加したメッセージ (Msg. x 1') の構造を、次のように定義し、上述のリーフ更新情報Msg. x 1を全てこのリーフ更新情報Msg. x 1' で置き替える。すなわち、リーフ更新情報Msg. x 1' は、

```
Leaf Entry Update Message {
    MessageID,
    FilteringMask,
    差分更新情報
}
```

このように定義される。「MessageID」(メッセージID)は、上述のリーフ更新情報Msg. x 1の場合と同様に、このメッセージ(リーフ更新情報Msg. x 1')の識別情報であって、例えば、このメッセージが生成される毎に1ずつ増加される整数である。「差分更新情報」は、ここに記述されるフィルタリングマスクで特定されるコンテナエントリ配下のリーフエントリの、追加、削除および属性変更といった手続の情報である。

【0101】

「FilteringMask」(フィルタリングマスク)の構造は、次のように定義される。すなわち、フィルタリングマスクは、

```
FilteringMask {
```

MaskSchema Version,

Mask Value

}

このように定義される。「MaskSchema Version」(マスキスキーマバージョン)は、例えば上述のコンテナ構造更新情報Msg. 1におけるメッセージIDに相当し、例えばこのフィルタリングマスクが生成される毎に1、増加される値である。「Mask Value」(マスク値)は、例えばビット列あるいはバイト単位で表されるマスクの値である。

#### 【0102】

なお、マスク値の構造は、マスキスキーマバージョンによって対応付けられるマスキスキーマ(後述する)によって規定される。マスキスキーマは、後述する別のメッセージによって送信側レプリケータ12から受信側レプリケータ17に通知される。

#### 【0103】

マスク値の割当方法について説明する。この実施の一形態では、あるコンテナエントリの配下のコンテナエントリのそれぞれを、所定ビット数からなるビット列で識別する。受信側レプリケータ17では、受信されたリーフ更新情報Msg. x1'中に記述されるマスク値を参照することによってフィルタ処理を行い、必要なリーフ更新情報Msg. x1'を選択的に抽出することができる。

#### 【0104】

フィルタリングマスクのマスク値のビット配列構造は、コンテナエントリの階層構造に対応させて決定される。例えば図17Aに一例が示されるように、図3で説明したエントリ名の記述方法に倣い、上位のコンテナエントリXの配下のエントリX.A、X.B、X.C、X.DおよびX.Eを互いに識別するために、それぞれ3ビットのマスク値(000)、(001)、(010)、(011)および(100)が割り当てられる。なお、[...]は、より上位のコンテナエントリが存在することを示す。

#### 【0105】

このようにマスク値が与えられた、コンテナエントリXの配下のコンテナエン

トリに対して、エントリの追加や削除が行われた場合、図 1 8 のフローチャートに従って処理が行われ、コンテナエントリの増減に応じてマスク値の割り当てを行う。なお、以下では、コンテナエントリの追加や削除が行われる前のコンテナ階層を、更新前コンテナ階層と称する。更新前コンテナ階層のマスク桁数  $M'$  は、送信側レプリケータ 1 2 の例えばメモリに記憶されているものとする。

## 【0 1 0 6】

先ず、最初のステップ S 6 0 で、送信側レプリケータ 1 2 によって、対象となるコンテナエントリの配下のコンテナエントリの数  $N$  が取得される。コンテナエントリ中の、配下のコンテナエントリのリストを参照することで、コンテナエントリ数  $N$  が求められる。次のステップ S 6 1 では、 $N$  個の要素を一意に識別可能なビット数  $M$  が選ばれ、マスクの桁数が  $M$  桁とされる。例えば、上述した図 1 7 A の例では、コンテナエントリ  $X$  は、配下のコンテナエントリを 5 個有しているので、5 個を一意に識別可能なビット数 [3] がマスクの桁数とされる。

## 【0 1 0 7】

次に、ステップ S 6 2 で、上述のステップ S 6 1 で割り当てられたビット数  $M$  が、対応する更新前コンテナ階層に割り当てられたマスク桁数  $M'$  と同一かどうか判断される。若し、マスク桁数  $M$  とマスク桁数  $M'$  とが同一であれば、処理はステップ S 6 3 に移行する。

## 【0 1 0 8】

ステップ S 6 3 では、更新後のコンテナ階層のコンテナエントリのうち、更新前コンテナ階層のコンテナエントリに対応するエントリには、同一のマスク値を割り当てる。さらに、次のステップ S 6 4 で、例えば更新後のコンテナ階層に新規にコンテナエントリの追加が起こったなどして、更新後のコンテナ階層に、更新前コンテナ階層に対応するコンテナエントリが存在しないコンテナエントリがあった場合、そのコンテナエントリに対して、同一のコンテナ階層の他のコンテナエントリのマスク値と重複しないようなマスク値が与えられる。

## 【0 1 0 9】

一方、上述のステップ S 6 2 で、マスク桁数  $M$  とマスク桁数  $M'$  とが同一ではないと判断されたら、処理はステップ S 6 5 に移行し、当該コンテナ階層の全コ

ンテナエントリのそれぞれに対して、一意にマスク値が与えられる。

【0 1 1 0】

例えば、上述の図 1 7 A の状態に、新たにコンテナエントリ "... X. F" が追加され、図 1 7 B のようなコンテナ階層になったとする。コンテナエントリ "... X" の配下のコンテナエントリ数  $N$  は、6 であり、一意に表現するためには 3 ビットが必要とされ、更新後のコンテナエントリ "... X" の配下のコンテナ階層のマスク桁数  $M = 3$  である。更新前コンテナ階層のマスク桁数  $M' = 3$  であって、マスク桁数  $M'$  とマスク桁数  $M$  とは等しい。したがって、図 1 7 B に示される各コンテナエントリ "... X. A"、"... X. B"、"... X. C"、"... X. D" および "... X. E" は、更新前コンテナ階層の対応するエントリのマスク値がそれぞれ割り当てられる（ステップ S 6 3）。一方、新規に追加されたコンテナエントリ "... X. F" は、同じコンテナ階層の他のコンテナエントリと重複しないように、マスク値 (1 0 1) が割り当てられる（ステップ S 6 4）。

【0 1 1 1】

また例えば、上述の図 1 7 A の状態から、コンテナエントリ "... X. C" を削除して、図 1 7 C のようなコンテナ階層になったとする。この場合、コンテナエントリ "... X" 配下のコンテナエントリ数  $N$  は、4 であり、2 ビットのマスク桁数  $M$  で各コンテナエントリを識別可能なので、更新後のコンテナ階層のマスク桁数  $M = 2$  である。一方、更新前コンテナ階層のマスク桁数  $M' = 3$  であって、更新前と更新後とで、マスク桁数が異なる。したがって、ステップ S 6 5 の処理によって、当該階層の全エントリに、マスク桁数  $M = 2$  で新たにマスク値が割り当てられる。

【0 1 1 2】

さらに、図 1 7 C の状態に、新たにコンテナエントリ "... X. G" が追加され、図 1 7 D の状態になったとする。この場合、コンテナエントリ "... X" 配下のコンテナエントリ数  $N$  は 5 であり、コンテナエントリを互いに識別するためには、マスク桁数  $M = 3$  とする必要がある。マスク桁数  $M$  が更新前のマスク桁数  $M' = 2$  と異なるため、ステップ S 6 5 の処理によって、コンテナエントリ "...

． X” の配下の全コンテナエントリに、新たにマスク値が割り当てられる。

【0113】

マスク値のビットアサインは、ディレクトリ構造の上位側から、コンテナ階層順にシリアルになされる。一方、この実施の一形態では、上述のように、同一階層に存在するエントリ数によってマスク桁数が異なる。また、エントリの削除や追加などによって、コンテナ階層中のエントリ数が変化し、それに伴いマスク桁数が変化する。そのため、マスク値を表すビット列中のどのビットがどのコンテナエントリ（あるいはコンテナ階層）に対応するかを判断し、マスク値を解釈するための情報機構が必要となる。

【0114】

この実施の一形態では、マスク値を解釈するための情報機構として、次に示すマスキューマ(MaskSchema)を定義する。マスキューマは、

```
MaskSchema {
    MaskSchema Version,
    TotalMaskLength,
    Set of ContainerEntryMaskSchema
}
```

このように定義される。「MaskSchema Version」(マスキューマバージョン)は、例えば上述のコンテナ構造更新情報Msg. 1におけるメッセージIDに相当し、例えば対応するフィルタリングマスクが生成される毎に1、増加される値である。「TotalMaskLength」(全マスク長)は、全体のコンテナ階層に対応する、マスク値全体のビット長を表す。すなわち、全マスク長は、ディレクトリ構造の全ての階層を表現するために必要なビット数に対応する。「Set of ContainerEntryMaskSchema」(セットオブコンテナエントリマスキューマ)は、後述する「ContainerEntryMaskSchema」(コンテナエントリマスキューマ)の配列を表す。

【0115】

上述のコンテナエントリマスキューマは、あるコンテナエントリに対応するフィルタリングマスクを規定する。すなわち、コンテナエントリマスキューマ

は、

```
ContainerEntryMaskSchema {
    ContainerEntryName,
    OffsetLength,
    MaskLength,
    AssignedMaskValue
}
```

このように定義される。「ContainerEntryName」（コンテナエントリ名）は、対象となるコンテナエントリのエントリ名を表す文字列である。「OffsetLength」（オフセット長）は、このコンテナエントリに対応するフィルタリングマスクの、全マスク値の最初のビットからのオフセット値であり、「MaskLength」（マスク長）は、マスク値の桁数（ビット長）である。「AssignedMaskValue」（割り当てマスク値）は、対象となるコンテナエントリに割り当てられたマスク値であり、ビット列で表される。

#### 【0 1 1 6】

図 1 9 を用いて、コンテナエントリマスクスキーマの符号化について説明する。図 1 9 A は、上述の図 1 7 A に対応する図であって、上位のコンテナエントリ “... X” の配下に、コンテナエントリ名 “... X. A”、“... X. B”、“... X. C”、“... X. D” および “... X. E” の 5 つのコンテナエントリが存在し、それぞれ 3 桁のマスク長で以てマスク値が割り当てられている。なお、ここでは説明のため、これら 5 つのコンテナエントリは、配下に他のエントリを有しないものとする。

#### 【0 1 1 7】

図 1 9 B は、コンテナエントリ “... X. C” のマスク値の一例を示す。この例では、オフセット長が 7 7 ビットであることから、コンテナエントリ “... X. C” に 3 ビットのマスク長で割り当てられた割り当てマスク値が、コンテナエントリ “... X. C” のマスク値の 7 8 ビット目から開始される 3 ビットであることが分かる。オフセット長に含まれる 7 7 ビットのマスク値は、コンテナエントリ “... X. C” より上位のコンテナエントリに対応する割り当てマスク値で



ある。

【 0 1 1 8 】

このように、マスク値における対象コンテナエントリの割り当てマスク値の位置が規定され、コンテナエントリマスクスキーマが符号化される。

【 0 1 1 9 】

コンテナエントリマスクスキーマのより具体的な例を示す。上述したコンテナエントリ "... X. C" に対応するコンテナエントリマスクスキーマは、例えば、

```
ContainerEntryMaskSchema {
    "...X.C", (ContainerEntryName)
    77,      (OffsetLength)
    3,       (MaskLength)
    010      (AssignedMaskValue)
}
```

このようになる。なお、括弧()内は、説明のためのものであって、実際に記述する必要は無い。

【 0 1 2 0 】

また、図 1 9 A に示されるコンテナエントリ "... X. D" に対応するコンテナエントリマスクスキーマは、例えば、

```
ContainerEntryMaskSchema {
    "...X.D",
    77,
    3,
    011
}
```

このようになる。

【 0 1 2 1 】

このときのマスクスキーマは、例えばマスクスキーマバージョンを 4 9 8、全マスク長を 1 3 4 ビットとした場合、

```

MaskSchema {
    498,  (MaskSchema Version)
    134,  (TotalMaskLength)
    ....
    ContainerEntryMaskSchema {
        "...X.C",
        77,
        3,
        010
    }
    ContainerEntryMaskSchema {
        "...X.D",
        77,
        3,
        011
    }
    ....
}

```

このようになる。上述の例では、コンテナエントリ "... X. C および "... X. D のコンテナエントリマスクスキーマがマスクスキーマ中に記述されているが、[...] の部分には、さらに他のコンテナエントリマスクスキーマが記述される。この例で分かるように、マスクスキーマには、一つのディレクトリ構造における全コンテナエントリに関するコンテナエントリマスクスキーマが記述される。

#### 【 0 1 2 2 】

なお、この例で、全マスク長が 1 3 4 ビットとなっているのに対して、コンテナエントリ "... X. C および "... X. D についてのコンテナエントリマスクスキーマでは、オフセット値が 7 7 ビットおよびマスク長が 3 ビットの、合計で 8 0 ビットである。これは、これらコンテナエントリ "... X. C および "...

X. Dの配下にも、さらにコンテナ階層が存在することを示している。

#### 【 0 1 2 3 】

上述したマスキーマにおいて、コンテナエントリ” ... X. Cに対応するフィルタリングマスクの符号化は、例えば、

```
FilteringMask {
    498,  (MaskSchema Version)
    .....010  (Mask Value)
}
```

このようになる。なお、マスク値(Mask Value)は、【 0 1 1 】以外の部分も全て、他の階層のコンテナエントリの割り当てマスク値からなるビットで埋められる。

#### 【 0 1 2 4 】

同様に、コンテナエントリ” ... X. Dに対応するフィルタリングマスクの符号化は、例えば、

```
FilteringMask {
    498,  (MaskSchema Version)
    .....011  (Mask Value)
}
```

このようになる。

#### 【 0 1 2 5 】

送信側レプリケータ 1 2 では、送信側サーバ 1 1 をモニタして、コンテナエントリの階層構造の変更を検知して、上述したマスキーマの変更を行う。したがって、受信側 3 において適切なフィルタ処理を行うためには、送信側レプリケータ 1 2 によって、階層構造の変更に基づく差分更新情報の通知と共に、変更されたマスキーマが受信側レプリケータ 1 7 に通知される必要がある。

#### 【 0 1 2 6 】

この発明では、マスキーマを送信側レプリケータ 1 2 から受信側レプリケータ 1 7 に通知するために、上述したコンテナ構造更新情報 M s g . 1 の構造に対して、マスキーマ構造を追加する。マスキーマ構造を追加されたコン

テナ構造更新情報Msg. 1' を、

Container Structure Update Message {

MessageID,

差分更新情報,

MaskSchema

}

このように定義する。マスキスキーマは、コンテナ階層の構成が変更される毎に変更される可能性がある。そのため、このコンテナ構造更新情報Msg. 1' も、コンテナ階層の構成の変更に応じて生成される。「MessageID」(メッセージID)は、コンテナ構造更新情報Msg. 1' が生成される毎に1ずつ増加される整数である。以下では、上述したコンテナ構造更新情報Msg. 1を、全てこのコンテナ構造更新情報Msg. 1' に置き替えるものとする。

#### 【0127】

送信側レプリケータ12では、上述した図15のフローチャートにおけるステップS46で、コンテナ階層に対応させたフィルタリングマスクが付加されたメッセージである、リーフ更新情報Msg. x1' を生成し、受信側レプリケータ17に放送している。ここで、受信側3では、リーフ更新情報Msg. x1' による受信側レプリケータ17でのフィルタ処理を行う前に、受信側クライアント15が必要としているコンテナ階層の対象部分を特定しておく必要がある。

#### 【0128】

この実施の一形態では、受信側レプリケータ17において、対象となるコンテナ階層をフィルタ処理するためのマスクをリストにした、ターゲットマスクリストを作成する。

#### 【0129】

図20を用いてターゲットマスクリストについて説明する。まず、図20Aに示されるようなディレクトリ構造を想定する。図20Aのディレクトリ階層は、最上位のルートエントリ以外は全てコンテナエントリで構成されているものとする。各四角はコンテナエントリを表し、二重線の四角で示されるコンテナエントリは、例えばユーザの嗜好に基づき受信側クライアント15でフィルタ処理を行

うように特定されたエントリである。各エントリ内に表示された数字は、当該エントリ毎に割り当てられたマスク値である。

#### 【0130】

図20Aに示されるように、ユーザの嗜好に基づきフィルタ処理するように受信側クライアント15で特定されたコンテナエントリのそれぞれに対して、マスク1～5が割り当てられている。このディレクトリ構造において、マスク1～5の全マスク長分のマスク値は、ディレクトリ構造を上位側から辿り、マスク1が〔000〕、マスク2が〔0010〕、マスク3が〔010〕、マスク4が〔10000〕およびマスク5が〔10010〕となる。

#### 【0131】

図20Bは、このように特定されたマスクがリストとされたターゲットマスクリストの一例を示す。ターゲットマスクリストは、ディレクトリの構造を特定するスキーマバージョンと、ユーザの嗜好などに基づき受信側クライアント15で特定された上述したマスク値のリストとからなる。すなわち、このターゲットマスクリストは、スキーマバージョンで記述されたディレクトリ構造でのみ有効なリストである。

#### 【0132】

図21は、ターゲットマスクリストを作成する処理のフローチャートである。このフローチャートは、受信側レプリケータ17で実行される。まず、最初のステップS70で、受信側レプリケータ17によって、コンテナ構造更新情報Msg. 1' が受信される。ステップS71で、ステップS70での受信がコンテナ構造更新情報Msg. 1' の初回の受信であるかどうか判断され、初回の受信であると判断されれば、処理はステップS73に移行する。

#### 【0133】

ステップS73では、受信されたコンテナ構造更新情報Msg. 1' のメッセージIDを、受信側レプリケータ17が有する、例えばメモリやハードディスクといった記録または記憶媒体に、コピー7として記憶される。

#### 【0134】

次のステップS74で、受信されたコンテナ構造更新情報Msg. 1' の内容

に基づき、コンテナ階層を生成する。生成されたコンテナ階層を示す情報が受信側レプリケータ 17 から受信側クライアント 15 に提示され、特定すべきコンテナエントリの選択が促される。例えば、受信側クライアント 15 では、所定の表示手段を用いて、供給されたコンテナ階層を示す情報に基づく表示を行う。ユーザは、この表示に基づき、所定の方法で必要なコンテナエントリを選択する。選択されたコンテナエントリの情報は、受信側クライアント 15 から受信側レプリケータ 17 に渡される。

## 【0135】

なお、コンテナエントリの特定は、ユーザの直接的な選択によってなされるのに限定されない。例えば、受信側クライアント 15 によって、ユーザが照会を行ったコンテナエントリの情報を蓄積し、蓄積された情報に基づきユーザの嗜好の傾向を学習して自動的に必要と思われるコンテナエントリを選択するようにもできる。さらに、ユーザによる直接的な選択と、学習による自動的な選択とを併用することもできる。

## 【0136】

このようにして、ステップ S 74 でコンテナエントリが選択されると、ステップ S 75 で、選択されたコンテナ階層に対応するフィルタリングマスクが設定される。設定されたフィルタリングマスクの一覧は、ターゲットマスクリストとして、例えば受信側レプリケータ 17 が有する、例えばメモリやハードディスクといった記録または記憶媒体に記憶される。

## 【0137】

一方、上述のステップ S 71 で、コンテナ構造更新情報 M s g. 1' の受信が初回ではないと判断されれば、処理はステップ S 72 に移行する。ステップ S 72 では、受信されたコンテナ構造更新情報 M s g. 1' のメッセージ ID が、前回までのコンテナ構造更新情報 M s g. 1' の受信によりステップ S 73 でコピー 7 として記憶媒体に記憶されたメッセージ ID と同一かどうか判断される。

## 【0138】

若し、両者が同一であると判断されたら、処理はステップ S 70 に戻される。一方、ステップ S 73 で両者が同一では無いと判断されたら、処理はステップ S

74に移行し、今回受信されたコンテナ構造更新情報Msg. 1' のメッセージIDが前回までのメッセージIDの代わりに記憶媒体に記憶され、今回受信されたコンテナ構造更新情報Msg. 1' に基づき以降の処理が行われる。

【0139】

図22は、図21のフローチャートに従って作成されたターゲットマスクリストに基づき、放送されたリーフ更新情報Msg. x1' を選択的に受信する処理を示すフローチャートである。受信側レプリケータ17は、ターゲットマスクリストに列挙されたフィルタリングマスクを有するリーフ更新情報Msg. x1' を、放送ネットワーク2で放送されたリーフ更新情報Msg. x1'の中から選択的に受信する。選択的に受信されたリーフ更新情報Msg. x1'により、上述した図14のフローチャートにおけるステップS32の処理が実行される。

【0140】

図22において、まず、最初のステップS80で、受信側レプリケータ17によって、放送ネットワーク2によって放送されたリーフ更新情報Msg. x1' が受信される。受信側レプリケータ17では、記憶媒体に記憶されているターゲットマスクリストが参照され、受信されたリーフ更新情報Msg. x1' に示されるフィルタリングマスクがターゲットマスクリストに存在するかどうか判断される。若し、ターゲットマスクリスト中に存在しなければ、処理はステップS80に戻る。

【0141】

一方、ステップS81で、当該フィルタリングマスクがターゲットマスクリスト中に存在すると判断されれば、処理はステップS82に移行し、ステップS80での受信がリーフ更新情報Msg. x1' の初回の受信であるかどうか判断される。若し、初回の受信であるとされれば、処理ステップS84へ移行し、受信されたリーフ更新情報Msg. x1' に示されるメッセージIDが、例えば受信側レプリケータ17が有する、例えばメモリやハードディスクといった記録または記憶媒体に、コピー8として記憶される。そして、次のステップS85で、受信されたリーフ更新情報Msg. x1' が受信側レプリケータ17での処理の対象として選択される。

## 【0142】

一方、ステップS82で、上述のステップS80でのリーフ更新情報Msg. x1'の受信が初回の受信では無いと判断されたら、処理はステップS83に移行する。ステップS83では、受信されたリーフ更新情報Msg. x1'のメッセージIDが、前回までのリーフ更新情報Msg. x1'の受信によりステップS84でコピー8として記憶媒体に記憶されたメッセージIDと同一かどうか判断される。

## 【0143】

若し、両者が同一であると判断されたら、処理はステップS80に戻される。一方、ステップS83で両者が同一では無いと判断されたら、処理はステップS84に移行し、今回受信されたコンテナ構造更新情報Msg. x1'のメッセージIDが前回までのメッセージIDの代わりに記憶媒体に記憶され、今回受信されたコンテナ構造更新情報Msg. x1'に基づき以降の処理が行われる。

## 【0144】

上述のようにして、受信側サーバ16で管理されるディレクトリ構造には、送信側サーバ11で管理されるディレクトリ構造のうち、受信側サーバ16を利用するユーザの嗜好を反映した部分のみを蓄積し、更新することができる。そのため、受信側サーバ16においてディレクトリ構造を蓄積するための蓄積記憶媒体を有効に利用することができる。また、それと共に、受信側サーバ16でのディレクトリ構造の格納コストを抑えることができる。さらに、受信側クライアント15による受信側サーバ16の内容の検索要求に対する処理効率を大幅に向上させることが可能となる。

## 【0145】

なお、上述では、各コンテナエントリに割り当てられた割り当てマスク長を可変長としたが、これはこの例に限定されない。割り当てマスク長は、例えばバイト単位などの固定長とすることもできる。

## 【0146】

## 【発明の効果】

以上説明したように、この発明によれば、受信側サーバで管理されるディレク



トリ構造には、送信側サーバで管理されるディレクトリ構造のうち、受信側サーバを利用するユーザの嗜好を反映した部分のみを蓄積し、更新することができる。そのため、受信側サーバにおいてディレクトリ構造を蓄積するための蓄積記憶媒体を有効に利用することができるという効果がある。

【0 1 4 7】

また、この発明によれば、受信側サーバでのディレクトリ構造の格納コストを抑えることができるという効果がある。

【0 1 4 8】

さらに、この発明によれば、受信側クライアントによる受信側サーバの内容の検索要求に対する処理効率を大幅に向上させることが可能となる効果がある。

【図面の簡単な説明】

【図 1】

この発明に適用できる系の一例を示す略線図である。

【図 2】

複数の受信側が放送ネットワークに接続されることを説明するための略線図である。

【図 3】

ディレクトリ構造を説明するための略線図である。

【図 4】

コンテナエントリの構造の一例を示す略線図である。

【図 5】

リーフエントリの構造の一例を示す略線図である。

【図 6】

送信側レプリケータの機能を説明するための機能ブロック図である。

【図 7】

受信側クライアントの機能を説明するための機能ブロック図である。

【図 8】

受信側サーバの機能を説明するための機能ブロック図である。

【図 9】

ディレクトリ構造の差分更新情報について説明するための略線図である。

【図 1 0】

ディレクトリ構造の差分更新情報について説明するための略線図である。

【図 1 1】

コンテナエントリの同期管理方法を説明するためのフローチャートである。

【図 1 2】

コンテナエントリの同期管理方法をより詳細に説明するためのフローチャートである。

【図 1 3】

コンテナエントリの同期管理方法をより詳細に説明するためのフローチャートである。

【図 1 4】

リーフエントリの同期管理方法を説明するためのフローチャートである。

【図 1 5】

リーフエントリの同期管理方法をより詳細に説明するためのフローチャートである。

【図 1 6】

リーフエントリの同期管理方法をより詳細に説明するためのフローチャートである。

【図 1 7】

フィルタリングマスクのマスク値のビット配列構造を説明するための略線図である。

【図 1 8】

エントリの追加や削除が行われた場合の、コンテナエントリの増減に応じたマスク値の割り当て処理のフローチャートである。

【図 1 9】

コンテナエントリマスクスキーマの符号化を説明するための略線図である。

【図 20】

ターゲットマスクリストを説明するための略線図である。

【図 21】

ターゲットマスクリストを作成する処理のフローチャートである。

【図 22】

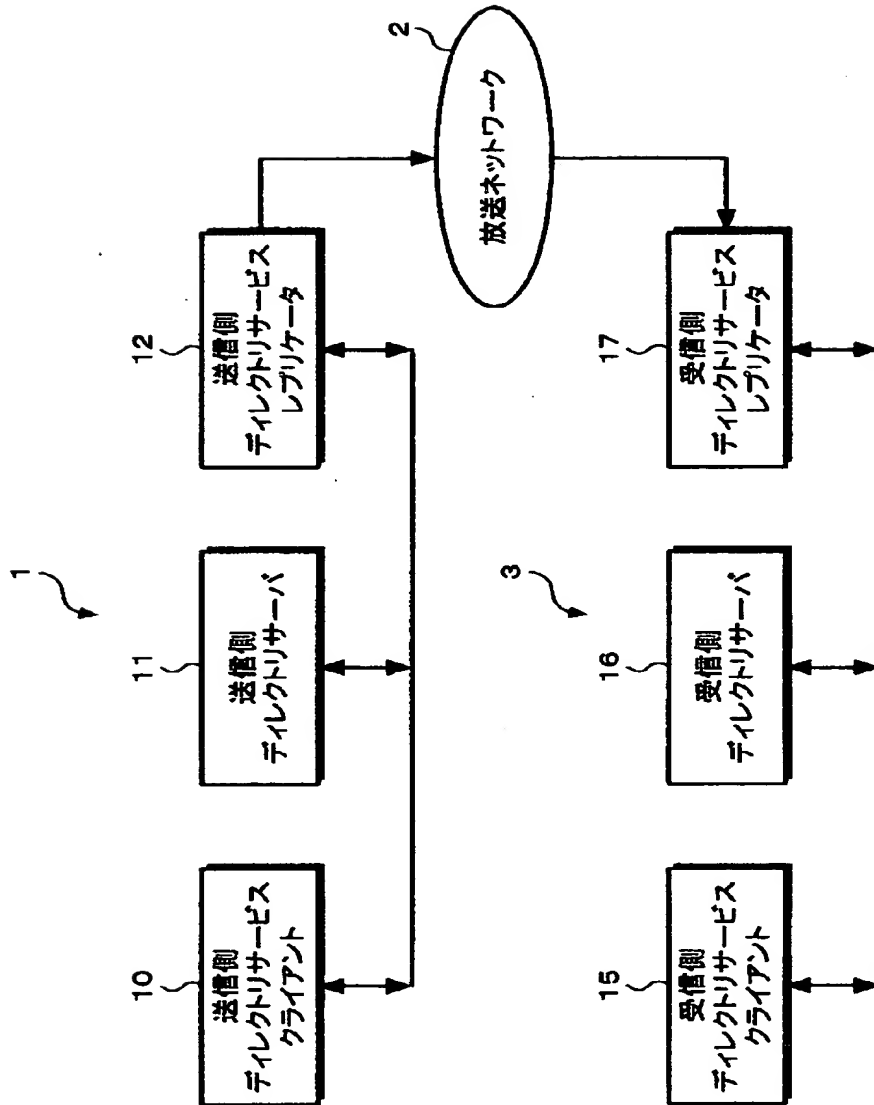
ターゲットマスクリストに基づき、放送されたリーフ更新情報Msg. x 1'を選択的に受信する処理を示すフローチャートである。

【符号の説明】

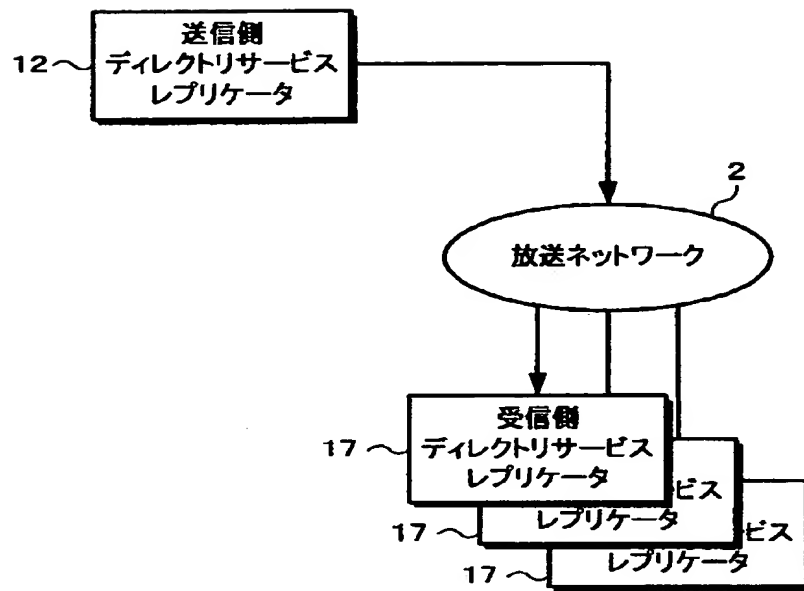
1・・・送信側、2・・・放送ネットワーク、3・・・受信側、10・・・送信側ディレクトリサービスクライアント、11・・・送信側ディレクトリサーバ、12・・・送信側ディレクトリサーバレプリケータ、15・・・受信側ディレクトリサービスクライアント、16・・・受信側ディレクトリサーバ、17・・・受信側ディレクトリサーバレプリケータ

【書類名】 図面

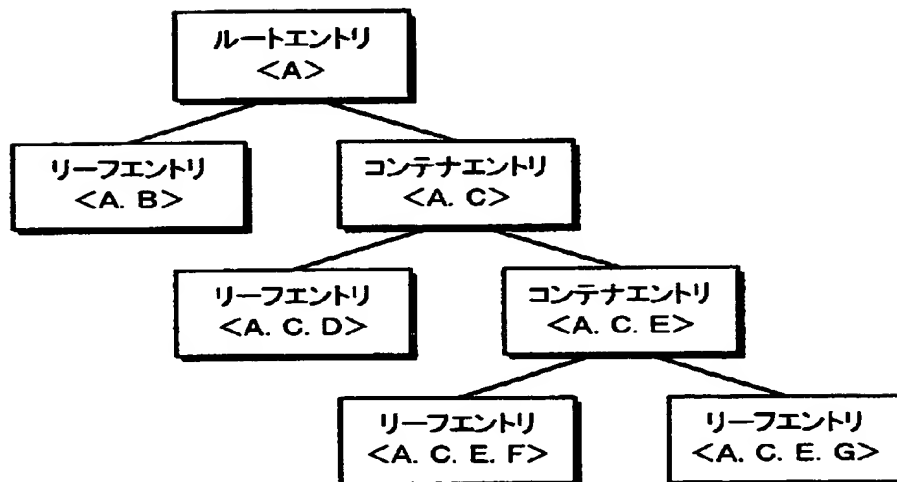
【図 1】



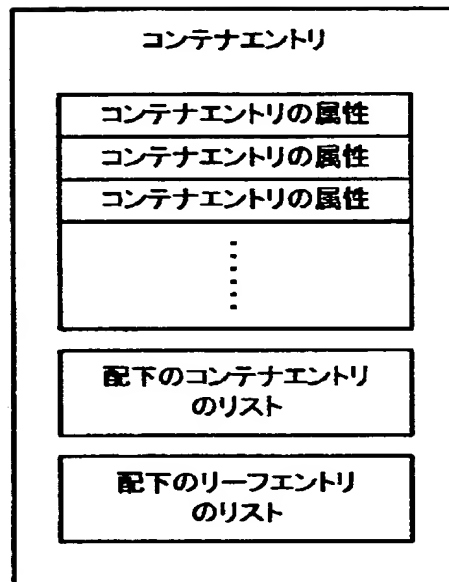
【図 2】



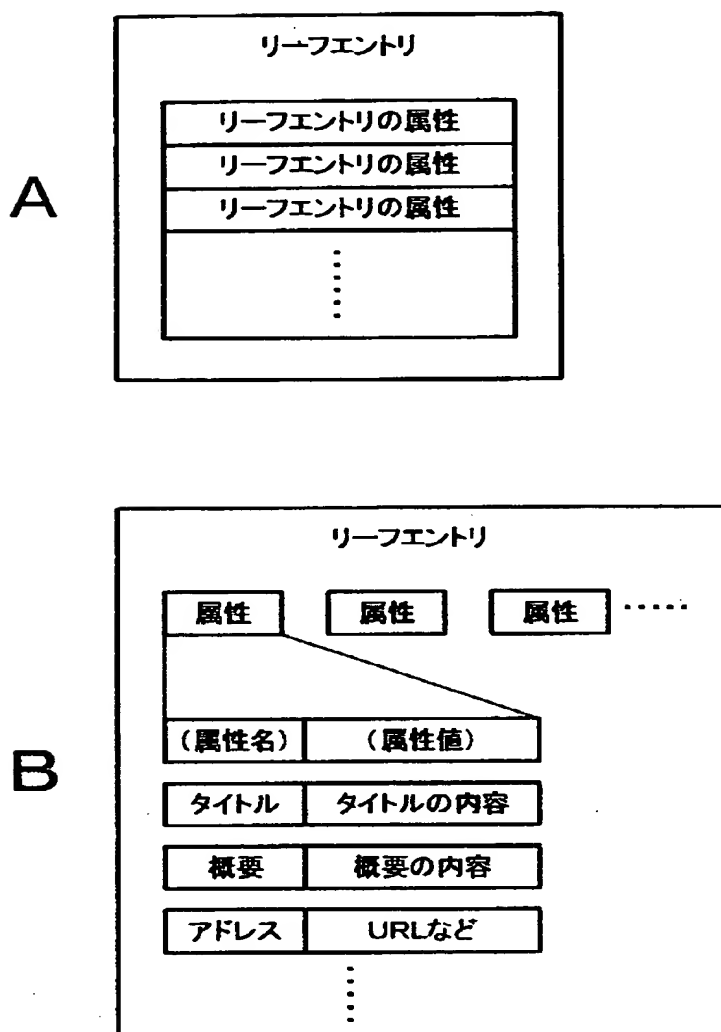
【図 3】



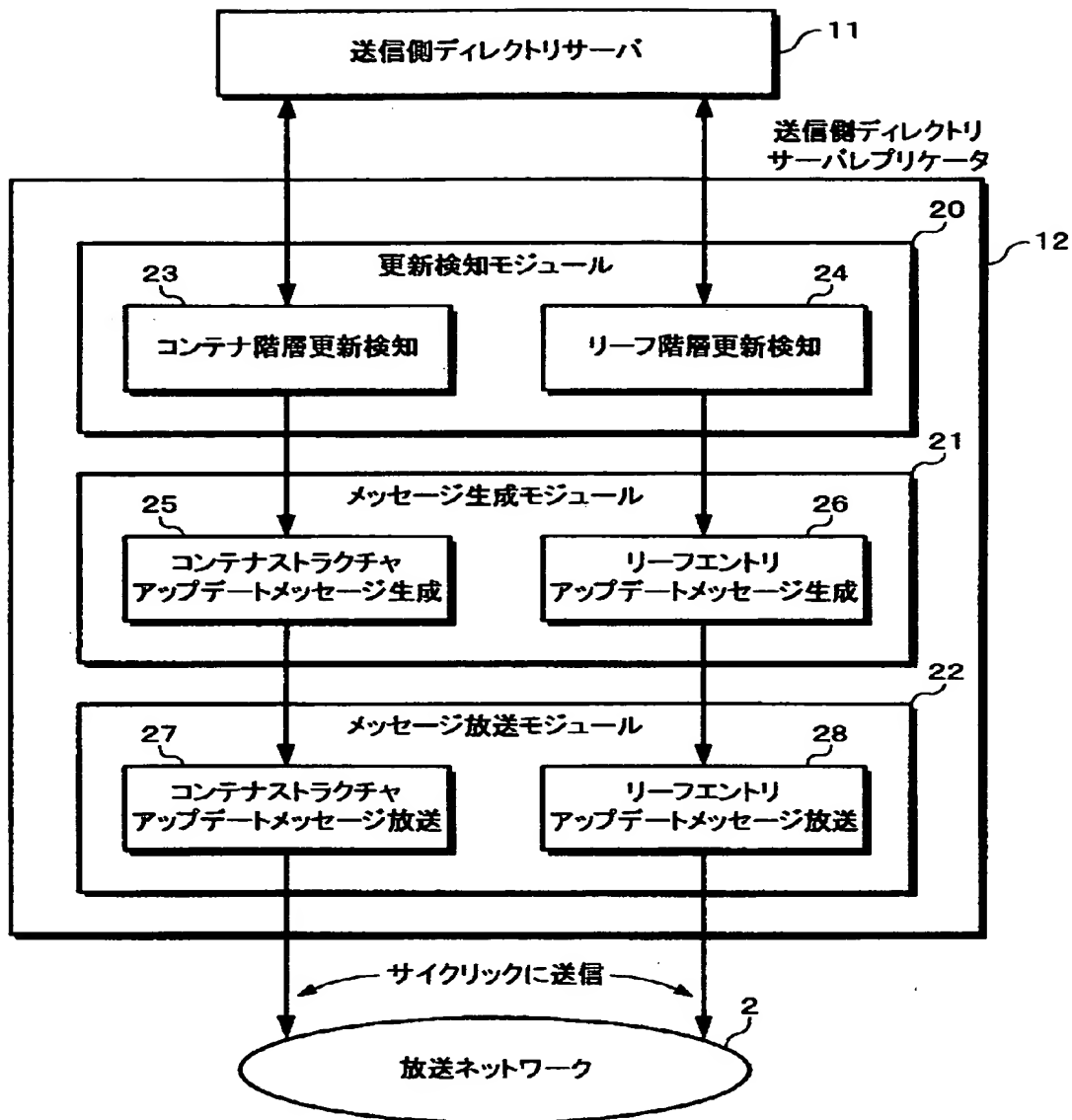
【図 4】



【図 5】

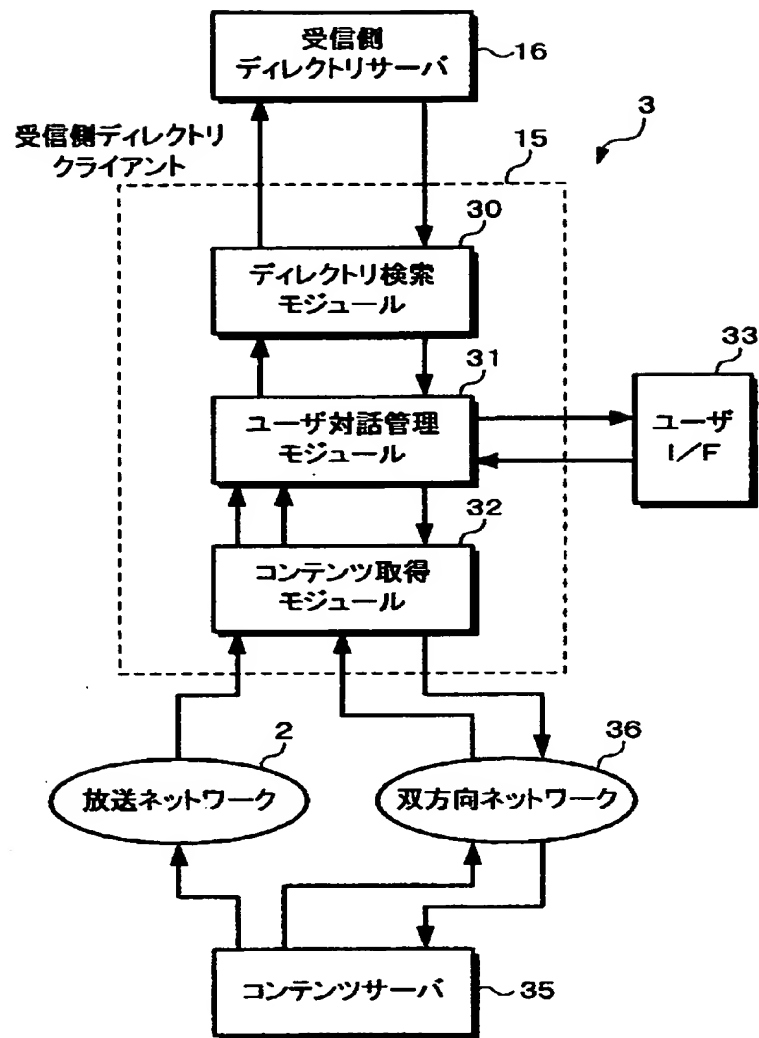


【図 6】

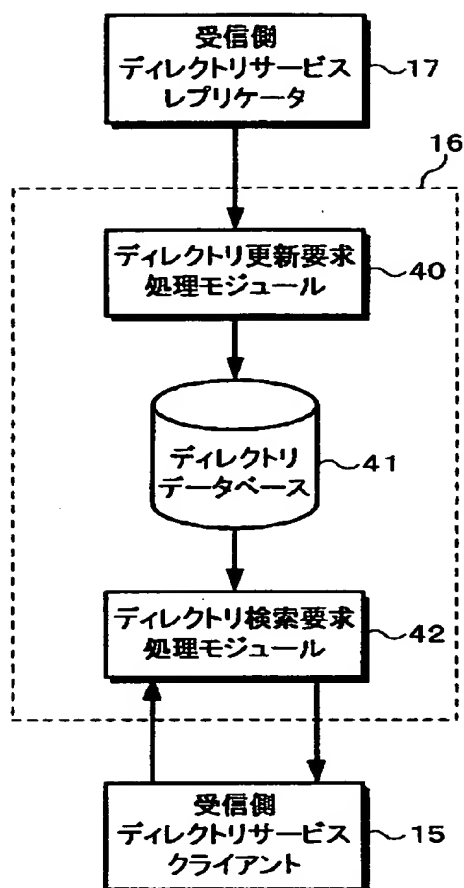




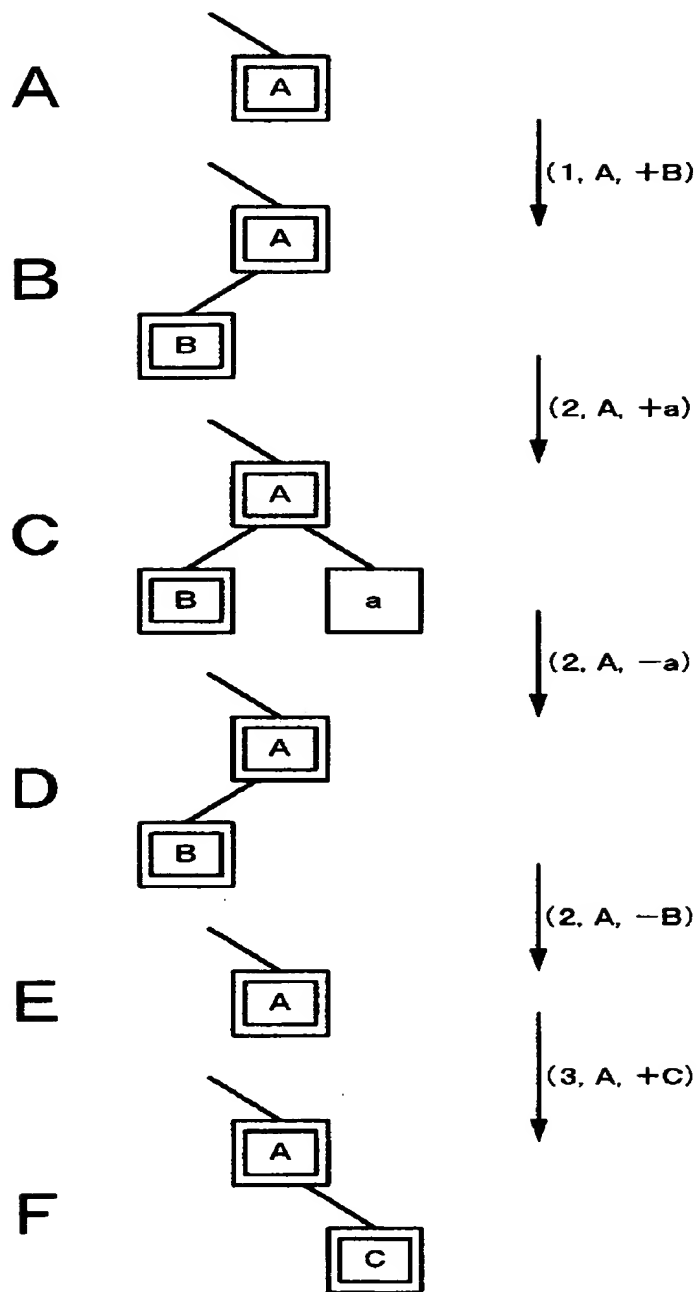
【図 7】



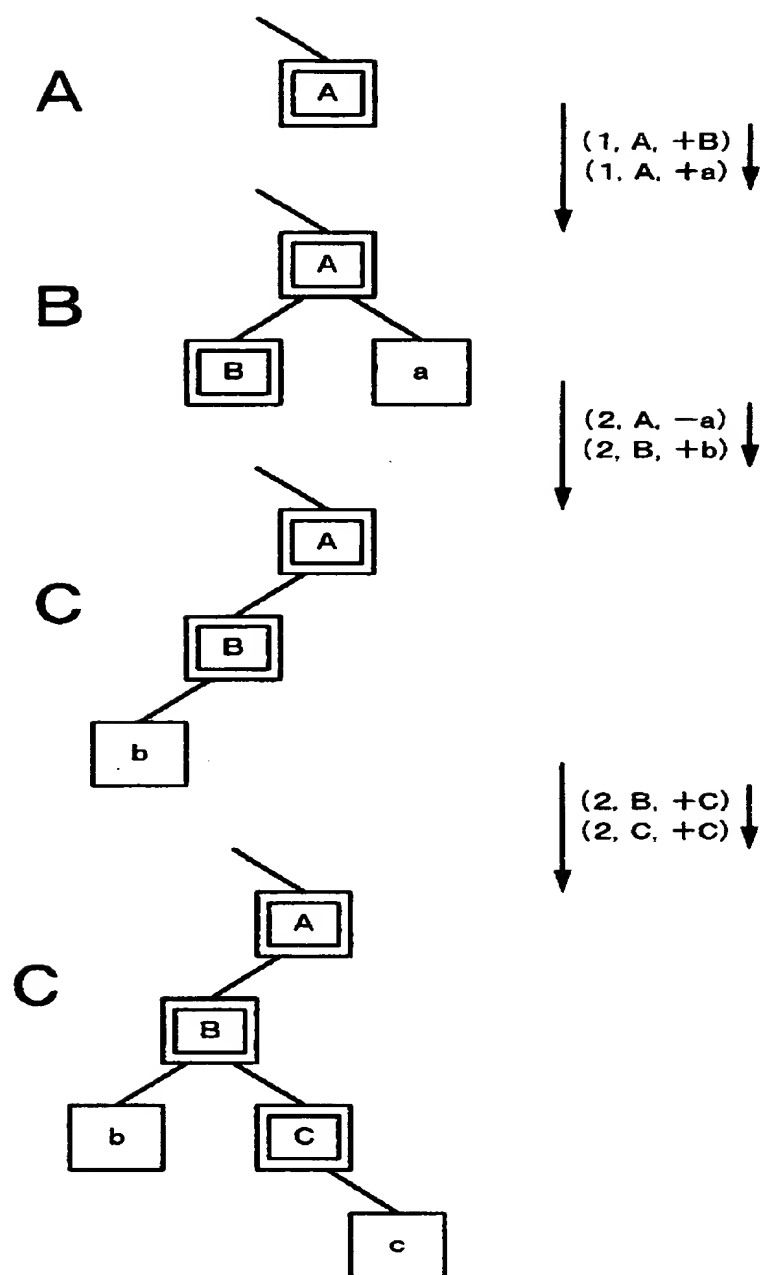
【図 8】



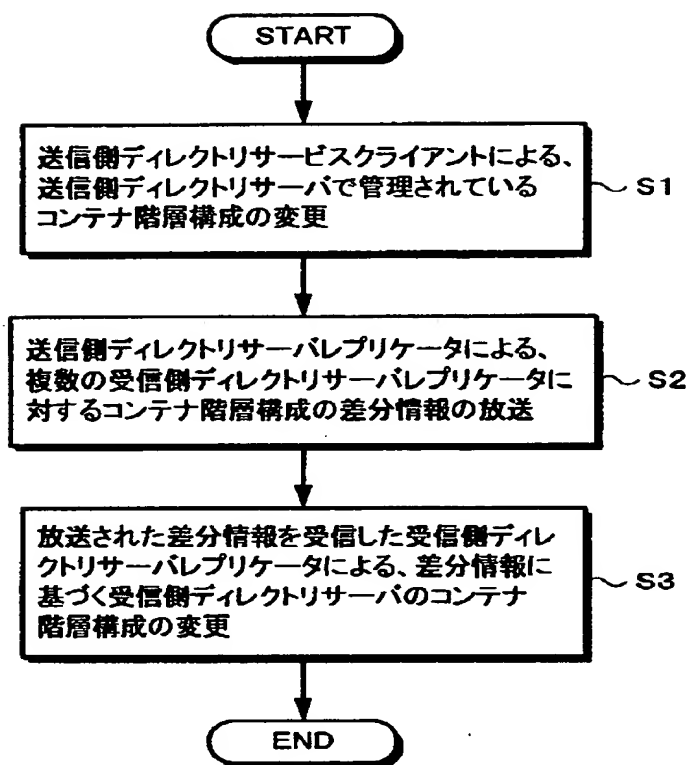
【図 9】



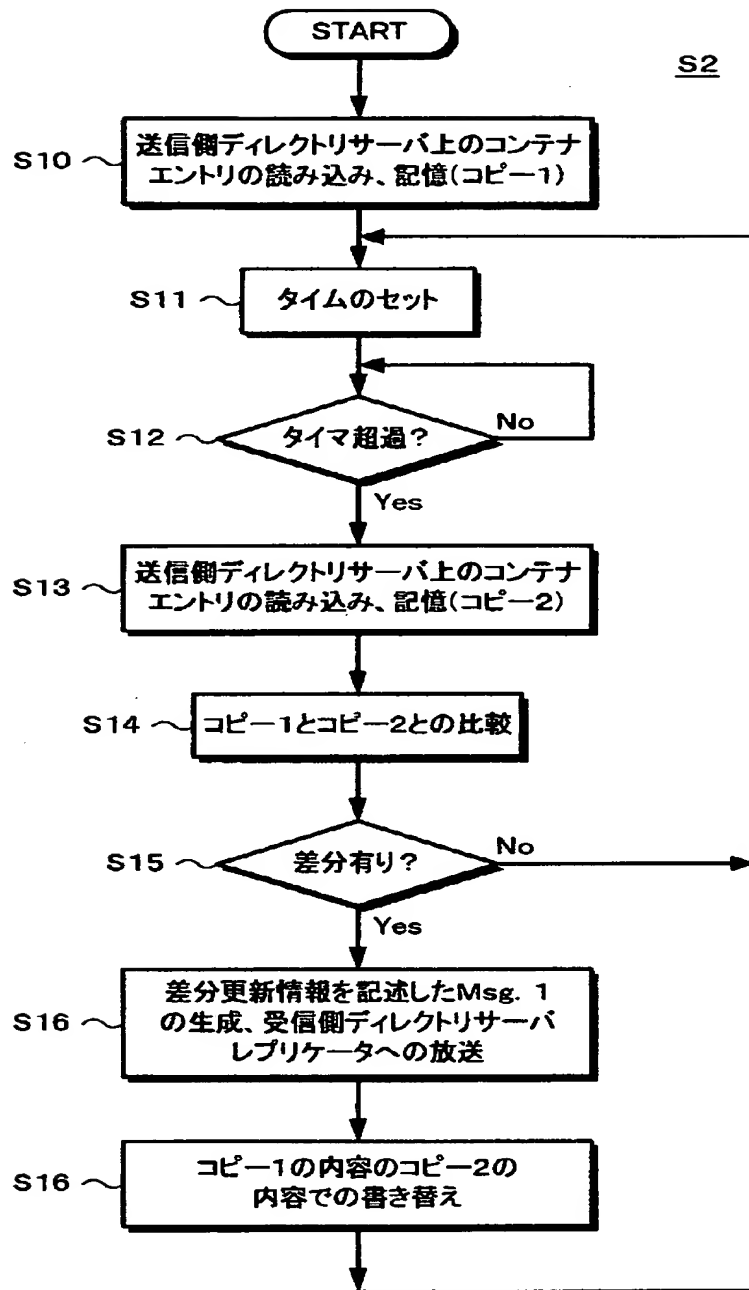
【図 1 0】



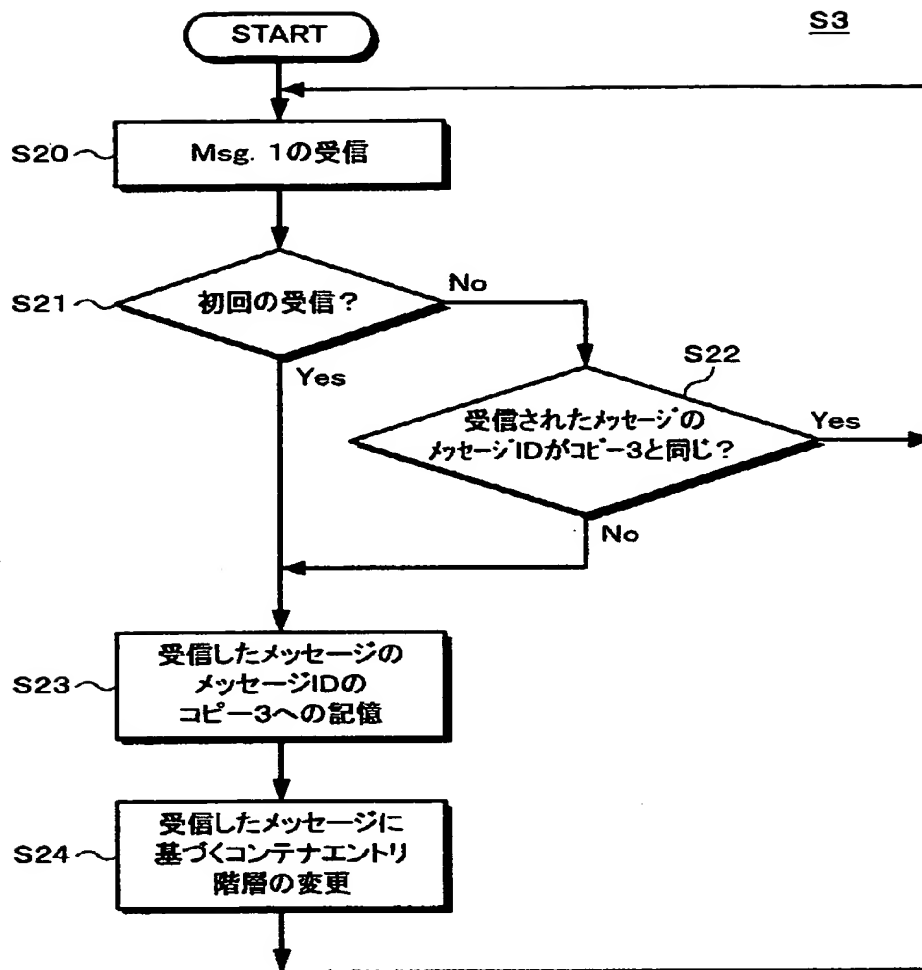
【図 1 1】



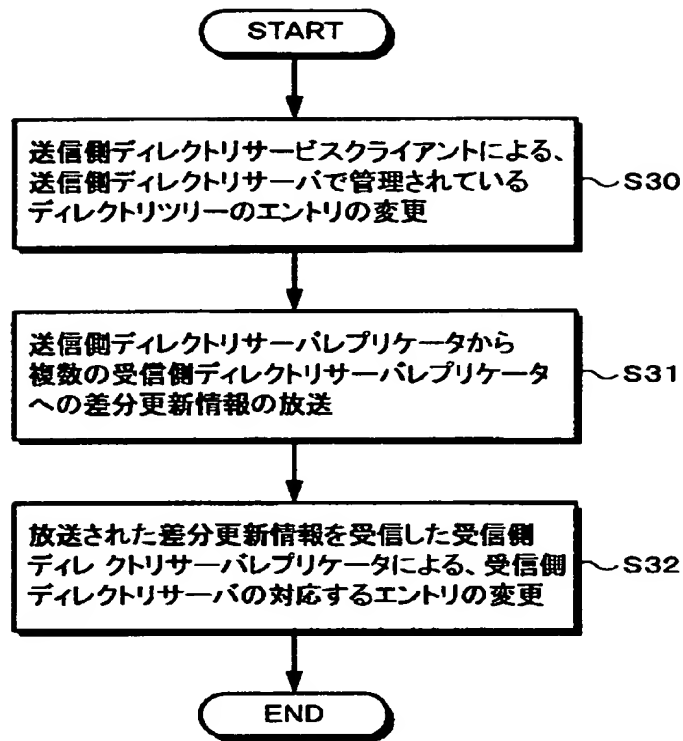
【図 1 2】



【図 1 3】

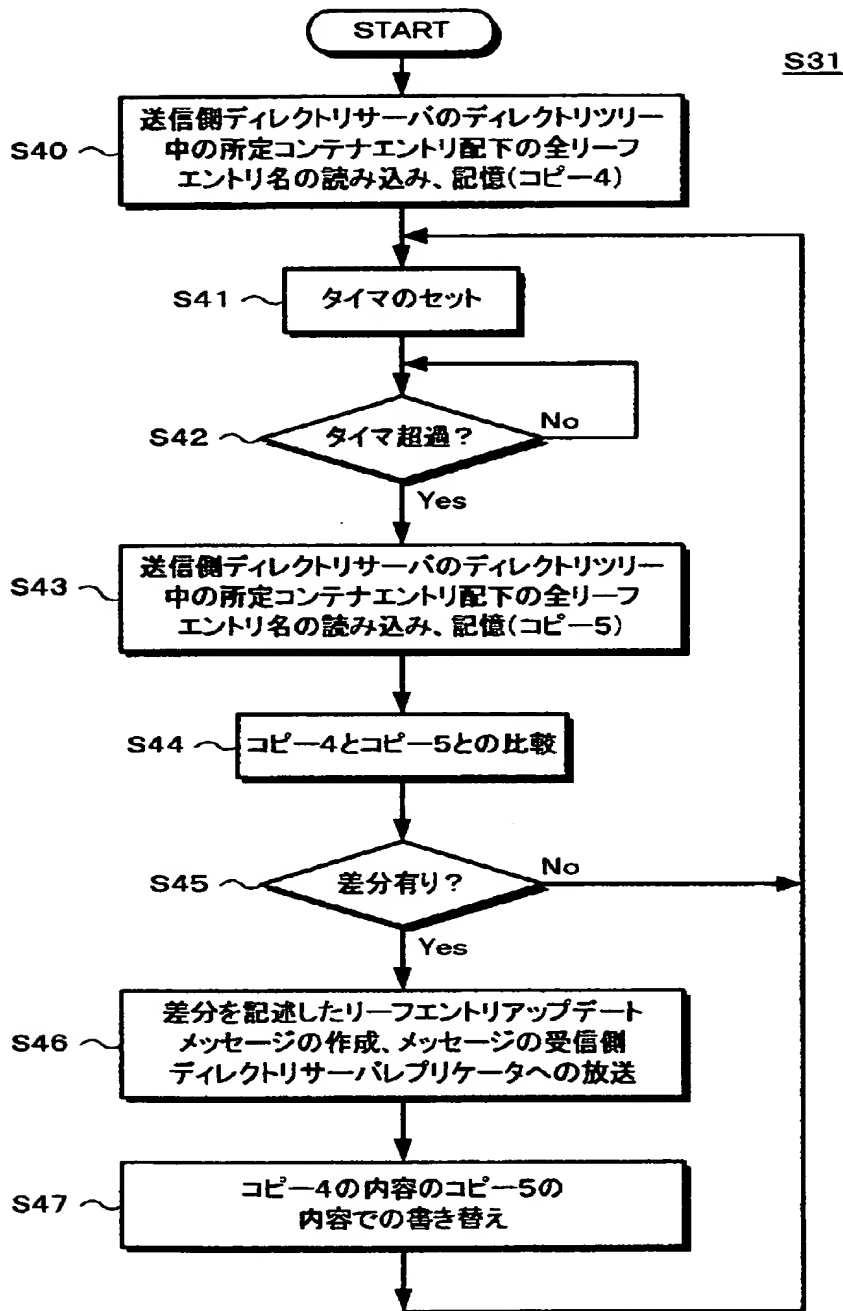


【図 14】

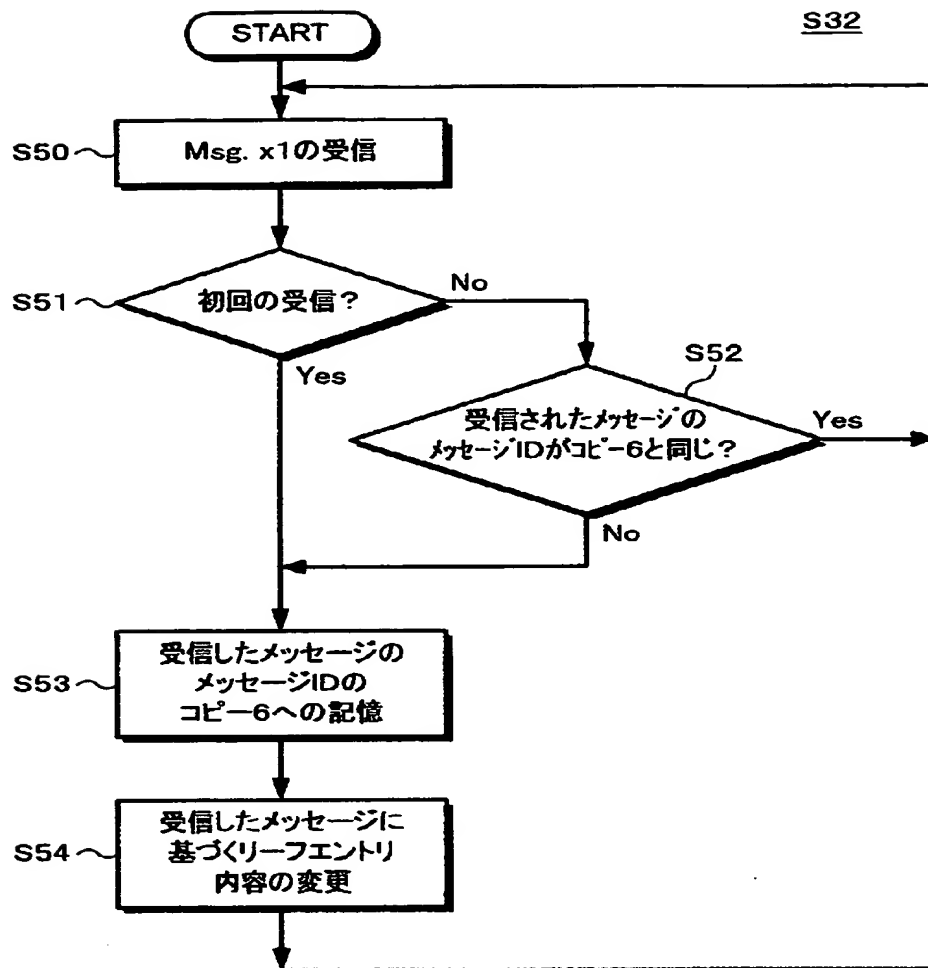




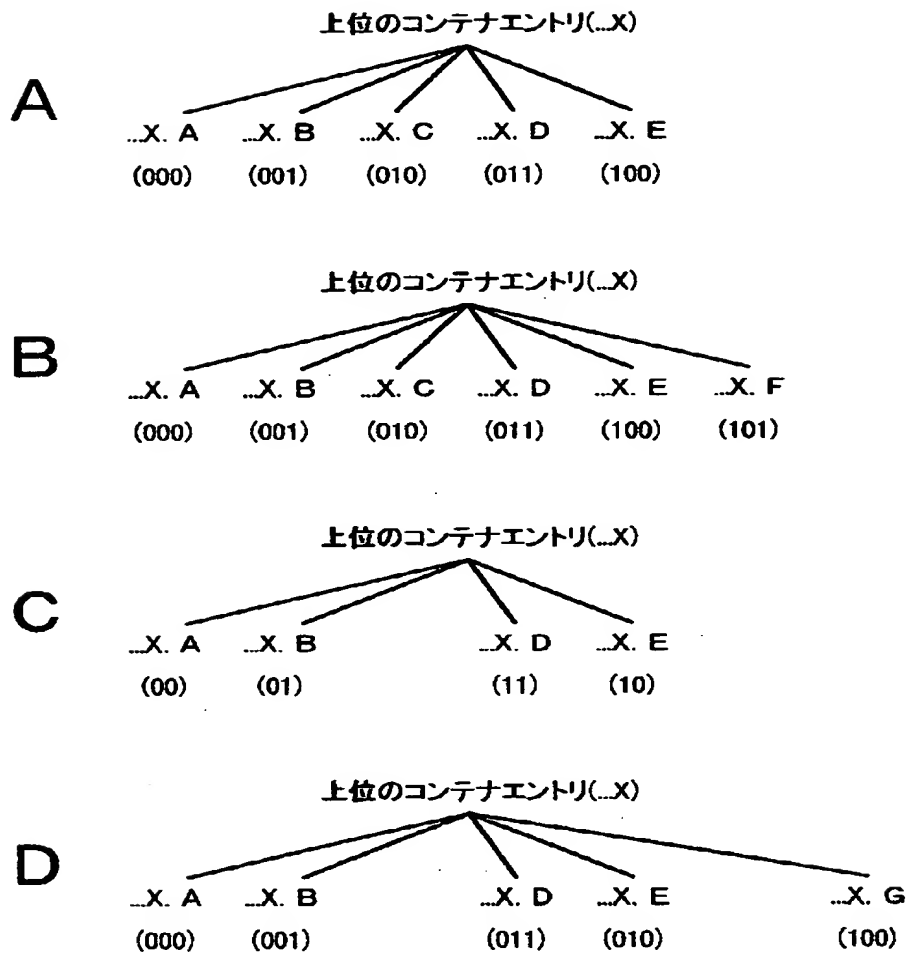
【図 1 5】



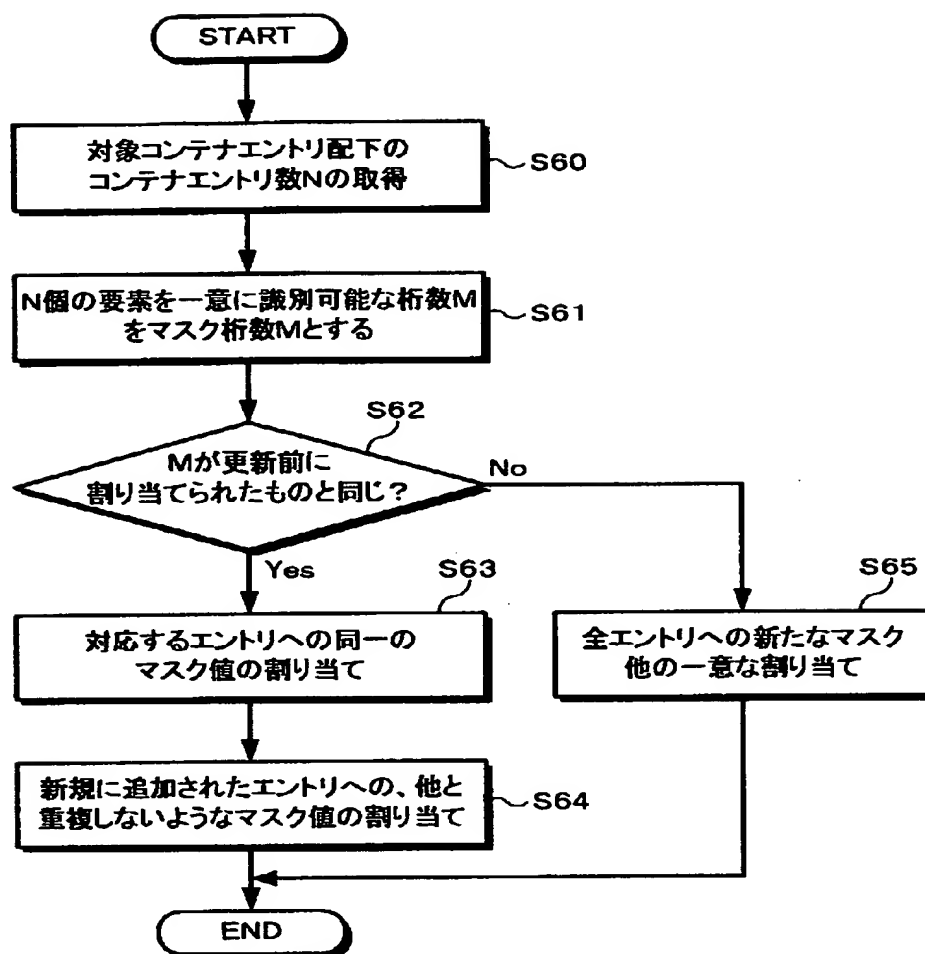
【図 1 6】



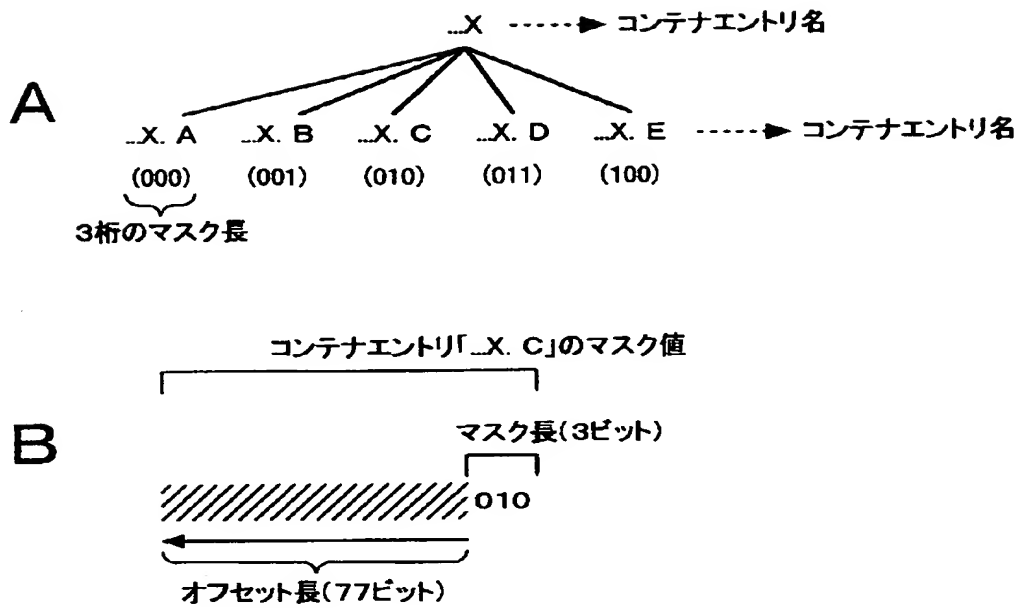
【図 1 7】



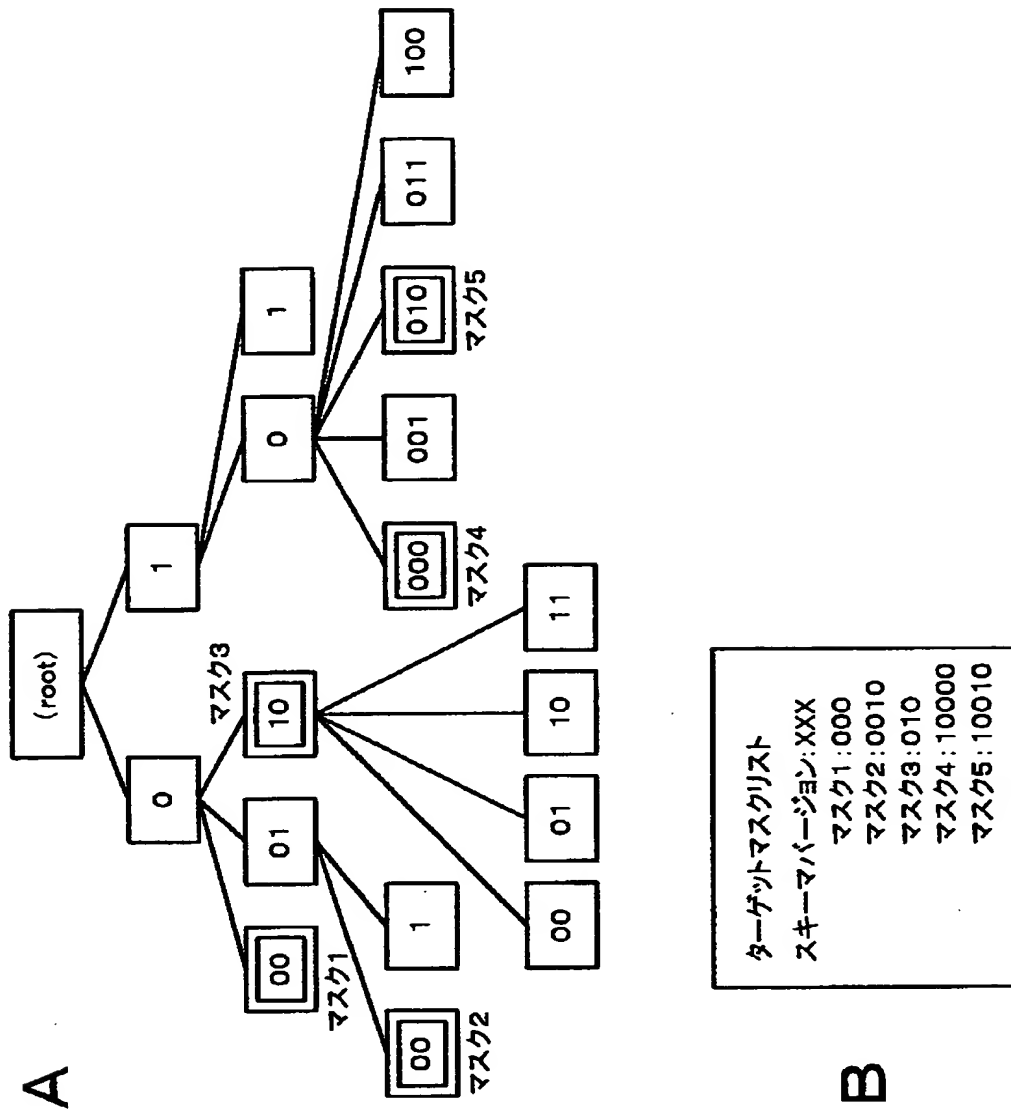
【図 18】



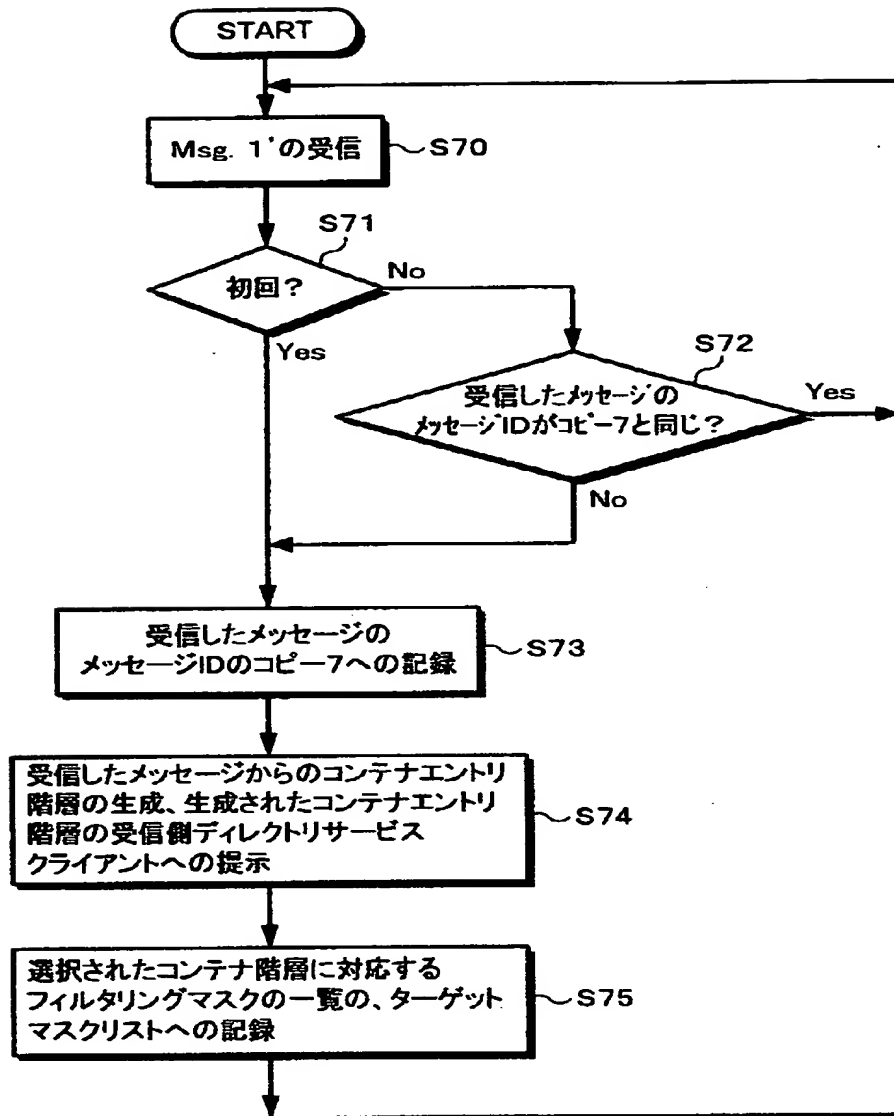
【図 1 9】



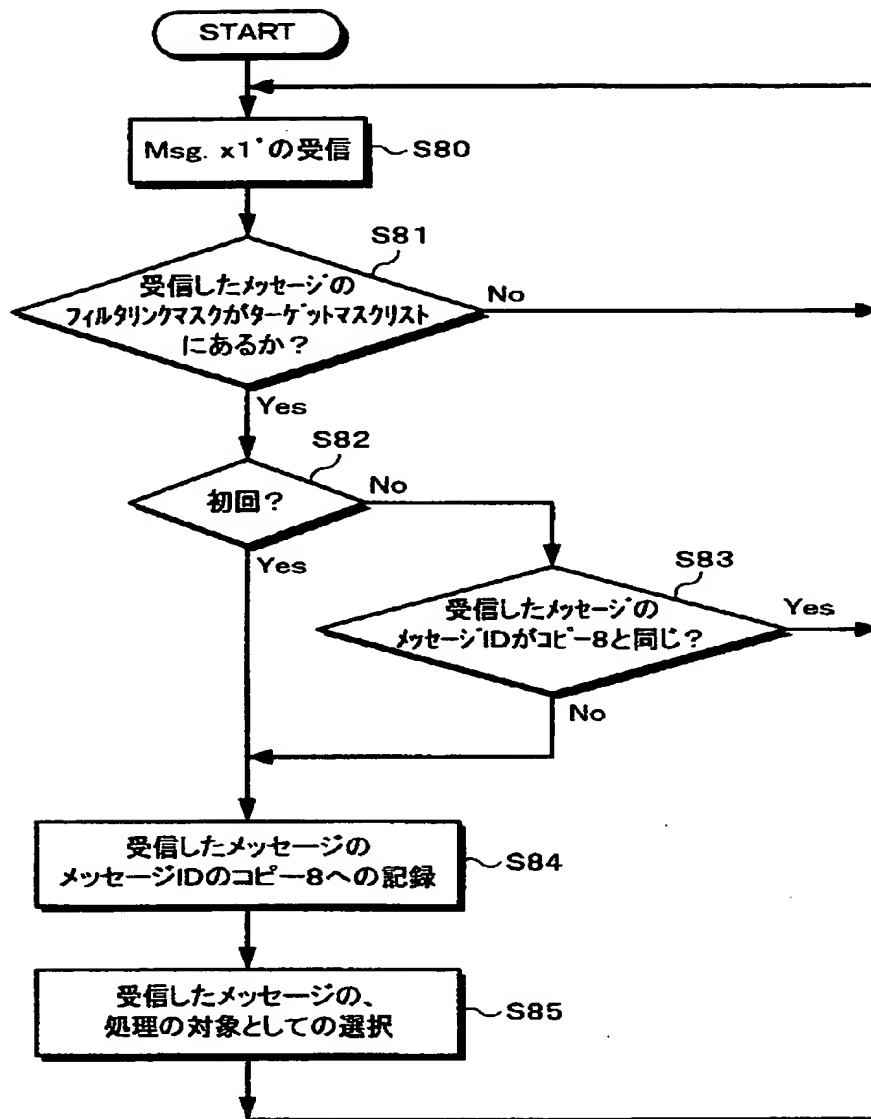
【図 2 0】



【図 21】



【図 2 2】





【書類名】 要約書

【要約】

【課題】 ディレクトリ階層構造の放送において、ユーザ側の負担が少なく、且つ、ディレクトリの階層構造の動的な変化に対応できるようにする。

【解決手段】 送信側 1 では、サーバ 11 に管理されているディレクトリ構造が変更されたことをレプリケータ 12 が検知する。検知結果に基づきコンテナ階層およびリーフ階層の差分更新情報がそれぞれ生成される。送信側 1 から放送されたコンテナ階層の差分更新情報が受信側 3 に受信され、クライアント 15 に提示され、所望のコンテナ階層の選択が促される。選択されたコンテナ階層に基づきコンテナ階層のマスクリストが作成される。送信側 1 から放送されたリーフ階層の差分更新情報は、受信側 1 で、コンテナ階層のマスクリストに基づきフィルタ処理され、選択的に取り込まれる。選択的に取り込まれたリーフ階層の差分更新情報によって、サーバ 16 で管理される受信側 1 でローカルなディレクトリ構造が更新される。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [ 5 9 7 1 3 6 7 6 6 ]

1. 変更年月日 1 9 9 7 年 9 月 2 6 日

[変更理由] 新規登録

住 所 東京都台東区西浅草 1 丁目 1 - 1

氏 名 株式会社次世代情報放送システム研究所

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社